

**PERFORMANCE EVALUATION OF IPv6 BGP
BASED SOLUTIONS FOR MALICIOUS ISP BLOCKING**

By

Mohammed Abdullaah AL-Mehdhar

A Thesis Presented to the

DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

Dhahran, Saudi Arabia

In Partial Fulfillment of the

Requirements for the Degree

MASTER OF SCIENCE

In

COMPUTER ENGINEERING

December, 2013

**PERFORMANCE EVALUATION OF IPv6 BGP BASED
SOLUTIONS FOR MALICIOUS ISP BLOCKING**

BY

Mohammed Abdullah Omer Al-Mehdhar

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

COMPUTER NETWORK

DECEMBER, 2013

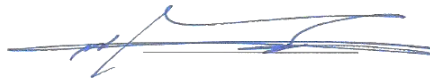
**KING FAHD UNIVERSITY OF PETROLEUM &
MINERALS**

DHAHRAN- 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **Mohammed Abdullah AL-Mehdhar** under the direction of his thesis advisor and approved by his thesis committee, has been presented and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN COMPUTER NETWORKS**.

Thesis Committee



Dr. Basem Al-Madani

Department Chairman

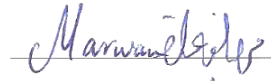


Dr. Salam A. Zummo

Dean of Graduate Studies

1/5/14

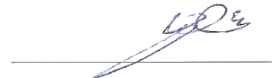
Date



Dr. Dr. Marwan Abu-Amara (Advisor)



Dr. Ashraf Hasan Mahmoud (Member)



Dr. Mohammed Houssaini Sqalli (Member)

Dedication

I am dedicating this thesis to four beloved people who have meant and continue to mean so much to me.

Mother .. Father.. Brothers .and. Sister

for their love, endless support and encouragement.

ACKNOWLEDGMENTS

First of all, Alhamdullillah to Allah S.W.T for giving me this opportunity and strength to complete my study. I also would like to thank the Hadramout Establishment for Human Development and King Fahd University of Petroleum and Minerals for supporting me in making my dream come true.

It has been customary to thank the supervisor for his effort in overseeing the progress of the thesis. I would like thank my supervisor, not only for this thesis, but also for helping me develop some vital skills within me. His intuitively driven scientific ideas, consistent support, motivation and encouragement have been the cornerstone of this research. I am also greatly appreciative of his valuable patience and time spent in discussing, editing and guiding during the write up of the thesis. A thank you note also goes to Dr. Ashraf Sharif Hasan Mahmoud and Dr. Mohammed Houssaini Sqalli for their great assistance, guidance and valuable time spent with me.

I would like also to express my deep thanks to the Computer Engineering department chairman, Dr. Basem Almadani, who offered unflagging support and wise advice.

I would like to thank the Hadhramout Establishment for Human Development, Yemen that provided the necessary financial support for this research. A special thanks go to Al sheikh Eng. Abdullah Ahmed Bugshan. I am just one amongst the many who have experienced his generosity and kindness. My words of thanks cannot repay what he has given me; yet, at this juncture I would like to record my statement of thanks to him for his kindness.

Finally, my heartfelt thanks, gratitude and appreciation goes to my family for their endless effort in persuading me to complete this thesis, and not forgetting my friends for their support and encouragement throughout the finishing point of this thesis.

TABLE OF CONTENTS

ACKNOWLEDGMENTS.....	iv
TABLE OF CONTENTS.....	vi
LIST OF TABLES	x
LIST OF FIGURES	xi
ABSTRACT	xv
ملخص الرسالة.....	xvii
CHAPTER 1 THE PROBLEM	1
1.1. Introduction	1
1.2. Motivation	3
1.3. Summary of Contributions	6
CHAPTER 2 BACKGROUND AND LITERATURE REVIEW	7
2.1. Background	7
2.2. Internet Protocol 6 (IPv6)	8
2.2.1. IPv6 Features and Benefits	8
2.3. Border Gateway Protocol (BGP)	11
2.3.1. BGP Attributes	11
2.3.2. BGP Path selection Procedure	14

2.3.3.	BGP Threats and Attacks	14
2.3.4.	BGP Security Solutions	16
2.3.5.	BGP Multihoming.....	17
CHAPTER 3 IMPLEMENTATION & VALIDATION OF SOLUTIONS USING BGP		
TUNING BASED APPROACH		18
3.1.	Introduction	18
3.2.	General Methodology	21
3.3.	BGP-Based Solutions	22
3.4.	Baseline Simulation Configuration	23
3.5.	Devices Used	23
3.6.	Controlling Traffic Using Supported OPNET features.....	25
3.6.1.	Baseline Simulation.....	25
3.6.2.	Outgoing Traffic Control Simulation	29
3.6.3.	Incoming Traffic Control Simulations.....	31
3.7.	Modification of OPNET Implementation	38
3.7.1.	OPNET Process Model	39
3.7.2.	Building A malicious Router	39
3.8.	Building Countermeasures Against a Malicious Act.....	45
3.8.1.	Summary of BGP in OPNET.....	45
3.8.2.	BGP Modification for Scheduling Reconfiguration.....	47
CHAPTER 4 PERFORMANCE EVALUATION OF BGP TUNING TECHNIQUES TO		
CIRCUMVENT MALICIOUS ACT		61
4.1.	Introduction	61
4.2.	Simulation Results and Analysis	63

4.2.1.	Percentage of traffic drop	63
4.2.2.	Convergence Time.....	66
4.2.3.	Throughput	68
4.3.	Comparison With IPv4.....	91
4.4.	Cases When The Simulation Fails	95
4.5.	Summary	96
 CHAPTER 5 CONCLUSION AND FUTURE WORK.....		97
5.1.	Conclusion	97
5.2.	Future Work.....	97
 Appendix A APPLICATION CONFIGURATION		99
A.1	TCP configuration	99
A.2	HTTP Configuration	100
A.3	FTP Configuration.....	101
A.4	VoIP Configuration	102
 APPENDIX B BASELINE THROUGHPUT		104
 Appendix C CODE MODIFICATION.....		108
C.1	ReconfigIn State	108
C.2	ReconfigOut State	112
C.3	Changes in BGP Module	116
C.4	Shortening	118
C.5	More Specific Prefixes	121
C.6	Modification in IP protocol.....	124

Appendix D FTP THROUGHPUT WITH INTER-REQUEST Time of 60 Seconds 128

References.....132

VITA135

LIST OF TABLES

Table 1.1 BGP ORIGIN Codes.....	13
---------------------------------	----

LIST OF FIGURES

Figure 1. 1 Malicious ISP blocking of the traffic in the region concerned.....	5
Figure 3. 1 Base Simulation Scenario.....	22
Figure 3. 2 Baseline Network Configuration.	23
Figure 3. 3 Incoming and Outgoing Traffic in Baseline Simulation.	26
Figure 3. 4 IP Forwarding Table for Router2.	27
Figure 3. 5 Convergence activity and duration of baseline Simulation.....	28
Figure 3. 6 IP forwarding table of Router5.....	28
Figure 3. 7 IP Forwarding Table of Router 2 After Applying Local-Preference Policy.	30
Figure 3. 8 Outgoing traffic from Router2.	30
Figure 3. 9 Convergence activity of Local-Preference Policy Scenario.....	31
Figure 3. 10 Incoming traffic to Router2 of prepending scenario.	32
Figure 3. 11 BGP routing table of Router3 in Prepend scenario.....	33
Figure 3. 12 Convergence activity of Prepending policy Scenario.	34
Figure 3. 13 BGP routing table of Router5 in Prepend scenario	34
Figure 3. 14 Incoming Traffic to Router2 Using Community.	36
Figure 3. 15 BGP Routing Table of Router5 in the Community Experiment.....	37
Figure 3. 16 Convergence Activity and Duration of Community Experiment.....	38
Figure 3. 17 IP_Dispatch Process Model.	40
Figure 3. 18 ip_rte_central_cpu Process Model.....	41
Figure 3. 19 Interface to Configure Malicious Router.....	43
Figure 3. 20 BGP table of Router 5 in Malicious Experiment.	44
Figure 3. 21 Throughput in A malicious Configuration Experiment.....	44
Figure 3. 22 bgp conn Process.....	46
Figure 3. 23 BGP Process.	46

Figure 3. 24 Modified BGP Process Model.	47
Figure 3. 25 Specification of Time When Applying Route Map.	48
Figure 3. 26 Throughput Traffic After Applying Local-Preference in the Presence of A malicious Router..	49
Figure 3. 27 Forwarding Table of Router2 in Local-Preference and Malicious Experiment.	50
Figure 3. 28 IP Forwarding Table of Router2 at Time 200 in the Local-Preference and Malicious Experiment.	51
Figure 3. 29 Routing Table of Router2 in Local-Preference and Malicious Experiment.....	52
Figure 3. 30 Throughput between Router2 and Router3, Router4 and Packet drop of Router3.	53
Figure 3. 31 BGP Routing table of Router5 in Malicious and Community Experiment.	54
Figure 3. 32 Convergence Activity in Community and Malicious.	55
Figure 3. 33 BGP Routing table of Router5 in Malicious and Community Experiment at Time 71 Seconds.	55
Figure 3. 34 BGP Routing Table of Router5 in Shortening, Local-Pref, Malicious Experiment.	57
Figure 3. 35 Throughput Between Router2 and Routers3 and Router4 and Dropped Traffic of Router3. ..	57
Figure 3. 36 BGP Routing Table of Router5 in As- Path Shortening and Local-Preference, Malicious Experiment.	58
Figure 3. 37 Convergence activity and duration for shortening experiment.	58
Figure 3. 38 Incoming and Outgoing Traffic of Router2 in More Specific , Local Preference, Malicious Experiment.	59
Figure 3. 39 Routing table for Router5 in More Specific Experiment.	60
Figure 3. 40 Convergence Activity and Duration of More Specific, Local Preference, and Malicious Experiment.	60
Figure 4. 1 Evaluation Network Setup.	61
Figure 4. 2 Network Showing the Links That will be Loaded With Traffic.	62
Figure 4. 3 Packet Drop Percentages.	64
Figure 4. 4 BGP Convergence Time for 0.1 Delay of Internet.....	66

Figure 4. 5 BGP convergence time for 5 second delay.	67
Figure 4. 6 Outgoing Throughput From Router2 to Router3 for HTTP.	69
Figure 4. 7 Throughput From Router2 to Router3 for FTP Application.	70
Figure 4. 8 Throughput From Router2 to Router3 for VoIP Application.	71
Figure 4. 9 Incoming Throughput to Router2 from Router3 for HTTP Application.....	72
Figure 4. 10 Throughput from Router3 to Router2 in FTP Application.	73
Figure 4. 11 Throughput from Router3 to Router2 for VOIP application.	74
Figure 4. 12 Outgoing Traffic From Router2 to Router4 for HTTP Application.	75
Figure 4. 13 Throughput from Router 2 to Router 4 for FTP Application.	77
Figure 4. 14 Throughput from Router2 to Router4 in VOIP Application.....	78
Figure 4. 15 Throughput from Router4 to Router2 for HTTP Application.....	79
Figure 4. 16 Throughput from Router4 to Router2 in FTP application.	80
Figure 4. 17 Throughput from Router4 to Router2 in VOIP Application.....	82
Figure 4. 18 HTTP Packet Sent by LAN_East.....	83
Figure 4. 19 HTTP Packet Received by LAN_East.....	84
Figure 4. 20 FTP Packets Sent from LAN_East.	85
Figure 4. 21 FTP Packet Received to LAN East.....	86
Figure 4. 22 VoIP Traffic Sent from LAN_East.	87
Figure 4. 23 VoIP Traffic Received by LAN_East.	88
Figure 4. 24 Page Response time for HTTP Client.	89
Figure 4. 25 FTP Download Response Time.	90
Figure 4. 26 Packets Drop in VoIP, More Specific solution, Exponential with 5 second delay, 80 link load.	95
Figure A. 1 TCP Configuration.	99
Figure A. 2 HTTP Configuration.	100
Figure A. 3 HTTP Page Properties.	100

Figure A. 4 Size of Image.....	101
Figure A. 5 HTTP Server Selection.	101
Figure A. 6 FTP Configuration.....	102
Figure A. 7 VoIP Configuration.	102
Figure A. 8 Talk Spurt Length.	103
Figure A. 9 Silence Length configuration.	103
Figure B. 1 Baseline HTTP Throughput from Router2 to Router3.	104
Figure B. 2 Baseline HTTP Throughput from Router2 to Router	105
Figure B. 3 Baseline Throughput from Router2 to Router3 for FTP traffic.	105
Figure B. 4 Baseline Throughput From Router3 to Router2 for FTP traffic.....	106
Figure B. 5 Baseline Throughput from Router2 to Router3 for VoIP traffic.....	106
Figure B. 6 Baseline Throughput From Router3 to Router2 for VoIP Traffic.	107

|

ABSTRACT

Full Name	[Mohammed Abdullah AL-Mehdhar]
Thesis Title	[Performance Evaluation of IPv6 BGP based Solutions for Malicious ISP Blocking]
Major Field	[Computer Networks]
Date of Degree	[December 2013]

The objective of this thesis is to evaluate several BGP-based techniques to overcome the intentional Internet blocking that is caused by a malicious Internet Service Provider (ISP) while assuming that the Internet is running IPv6. We evaluate the BGP-based solutions while controlling the incoming and outgoing traffic through a non-malicious ISP. The first contribution of this thesis is the implementation of the problem model and the BGP tuning methods using OPNET for IPv6 networks. The implemented methods are AS-Path, shortening, more specific prefixes, using of community, and local-preference. The second contribution is evaluating the BGP based solutions by discussing packet drop, convergence time, links throughput and application throughput. Based on the results obtained, the more specific prefix method has the lowest convergence time while the shortening and community methods have almost the same convergence time. However, the community method has the lowest dropped packets percentage. All methods have almost the same performance for the throughput. The third contribution is performing a comparison between our results and the IPv4 results

obtained by Alrefai [1]. Based on the results obtained, the more specific prefix method has the lowest convergence time while the shortening and community methods have almost the same convergence time. However, the community method has the lowest dropped packets percentage. All methods have almost the same performance for the throughput. Finally, the results of the performance evaluation were compared against the results obtained by Alrefai [1].

ملخص الرسالة

الاسم الكامل: محمد عبدالله عمر المحضار

عنوان الرسالة: تقييم الأداء للحلول المعتمدة على بروتوكول بوابة الحدود (BGP) لمشكلة الحجب من مزود

الانترنت الدولي لبروتوكول الانترنت IPv6

التخصص: شبكات

تاريخ الدرجة العلمية: ديسمبر 2013

الهدف من هذه الرسالة هو تقييم الحلول المقدمة من الباحث الرفاعي [1] المعتمدة على بروتوكول بوابة الحدود (BGP) لحل مشكلة حجب الانترنت عن منطقة محلية باستخدام بروتوكول الجيل الثاني من بروتوكول الانترنت (IPv6). هذه الدراسة تهتم بالحجب المتعمد من قبل مزود الانترنت الخبيث للمنطقة المحلية مع استمراره الاعلان عن وجود مسارات عبرة للمنطقة المحلية. قمنا بتقييم حلول البوابة الحدودية بالتحكم بالحزم الصادرة والواردة للمنطقة المحلية عبر مزود اخر غير خبيث. قمنا بعمل نموذج محاكاة باستخدام برنامج الـ (OPNET) لبروتوكول الانترنت (IPv6). كذلك قمنا بتقييم أداء هذه الحلول من حيث الحزم المسقطة، وقت التقارب، والانتاجية، بالإضافة لبعض القياسات الخاصة بتطبيقات محددة. وفي الاخير قمنا بمقارنة النتائج المتحصل عليها من تجاربنا مع النتائج التي حصل عليها الباحث الرفاعي [1]

CHAPTER 1

THE PROBLEM

1.1. Introduction

The Internet has become part of modern human civilization, and the influence of the Internet has touched every aspect of life. The way information is retrieved has also changed drastically, as news and information can be accessed instantly from anywhere in the world through the Internet. Most of the conventional services have been changed to match the Internet environment. TV and radio, business applications that give new services such as e-shopping, banking services and e-governments, have all adapted to the Internet.

In this respect, the Internet provides more sophisticated services such as voice over IP, video calls and instant messaging services, which have paved the way for interactive websites such as social networking sites. All these network services demonstrate the demand for communication infrastructure that will guarantee their stability and availability.

Reaching any of these services requires the user to have Internet access. Home, office, university or public access points are different types of networks that users can utilize to connect to the Internet. When a user connects to the Internet through an Internet Service Provider (ISP), that user becomes part of the ISP network. This network in turn connects to a larger network and becomes part of that network and so on. The Internet is basically a network of networks. A network that works under a single

administration is referred to as an Autonomous System (AS). An AS could be a local or an international ISP (IISP).

When information is sent or received over the Internet, it actually moves from the user's ISP network to another ISP network until it reaches its final destination. The routing process in the Internet can be classified into two phases. The first phase is an internal phase, where the routing process controls the traffic inside an AS. The second phase is the exterior routing process, where it controls the traffic between different ASes. The later phase is done through Border Gateway Protocol (BGP), which is the inter-AS routing protocol of the Internet. BGP selects the best path using a combination of different rules [2]. Accordingly, the selected path is not necessarily the shortest path, but rather the best that matches the ASes routing policies. Thus, the BGP routing is called policy-based routing [3].

For the Internet routing to proceed, an Internet Protocol (IP) address is needed. The currently deployed version of IP addresses is IP version 4 (IPv4). However, due to the growing demands for Internet access, IPv4 addresses are being very rapidly depleted. To address this problem, different types of solutions have been proposed such as Network Address Translation (NAT). Another proposed solution is IP version 6 (IPv6) which is considered to be the next generation of Internet protocols. IPv6 solves the IPv4 address limitation by increasing the IP address from 32 bits to 128 bits. In addition, IPv6 has more new and enhanced features such as enhanced security, new flexible header and fast handover [4].

Due to the high dependency on the Internet for all aspects of our lives, the Internet needs to be stable and resilient. Disasters, human mistakes or malicious behaviors are examples of causes of Internet outage which can result in different levels of damage.

In general, outage causes can be classified into deliberate and non-deliberate. Depending on the type and the level of outage, an AS that experiences an Internet outage becomes isolated from the entire Internet or from parts of it.

This study aims to evaluate several BGP-based techniques to overcome the intentional Internet blocking caused by a malicious ISP while assuming that the Internet is running IPv6. None of the new features of IPv6 BGP have been used in this study. These techniques are evaluated in a simulation environment to assess their performance with respect to convergence time and effect on Internet applications.

1.2. Motivation

As IPv6 is expected to be deployed soon in Saudi Arabia, the Internet resiliency against outages is one of the main concerns. However, Internet resiliency can be compromised and Internet outage may happen as a result of different types of malicious activities. Thus, in order to prevent an outage and to provide a higher level of Internet resiliency, the outage causes must be investigated and preventive and recovery solutions should be proposed. In this study, we consider IPv6 and BGP-4 attributes and methods as possible solutions to providing higher level of Internet resiliency. Problem Description

An Internet outage may occur because of deliberate or non-deliberate reasons, and can result in different levels of blocking. The target of the blocking can be at the hardware level or at the software level. The hardware level blocking includes link or router blocking. On the other hand, the software level blocking may be divided into network level and application level. At the network level, a change in the BGP configuration could lead to blocking a specific region. On the other hand, an application level blocking can occur by falsifying DNS messages or denying access to the DNS service.

The most effective and widely used method to cause an Internet blocking is the network level blocking that can be achieved by blocking IP traffic at the network layer. Access Control List (ACL) commands can be used to achieve incoming and outgoing traffic blocking for a specific IP address. In this work, we consider the blocking of IP traffic at the network layer.

In general, the blocking of IP traffic by a malicious ISP happens when two conditions are met:

1. The traffic goes through the malicious ISP's network.
2. The malicious ISP drops packets that carry the targeted source or destination IP addresses.

Hence, the blocking of IP traffic by a malicious ISP problem can be resolved by eliminating one or both of these conditions. Subsequently, two classes of solutions can be considered: Solutions to control the traffic path, so that it does not pass through the

malicious ISP; and solutions to prevent traffic from being dropped at the malicious ISP by concealing the traffic identity.

This study focuses on evaluating the performance of the BGP solutions for the blackholing of the traffic originating from or destined for the local region that is caused by the malicious ISP. The solutions considered by this study are based on controlling traffic through the use of BGP over IPv6. |

Specifically, Figure 1.1 shows the four parts of the studied network: local region,

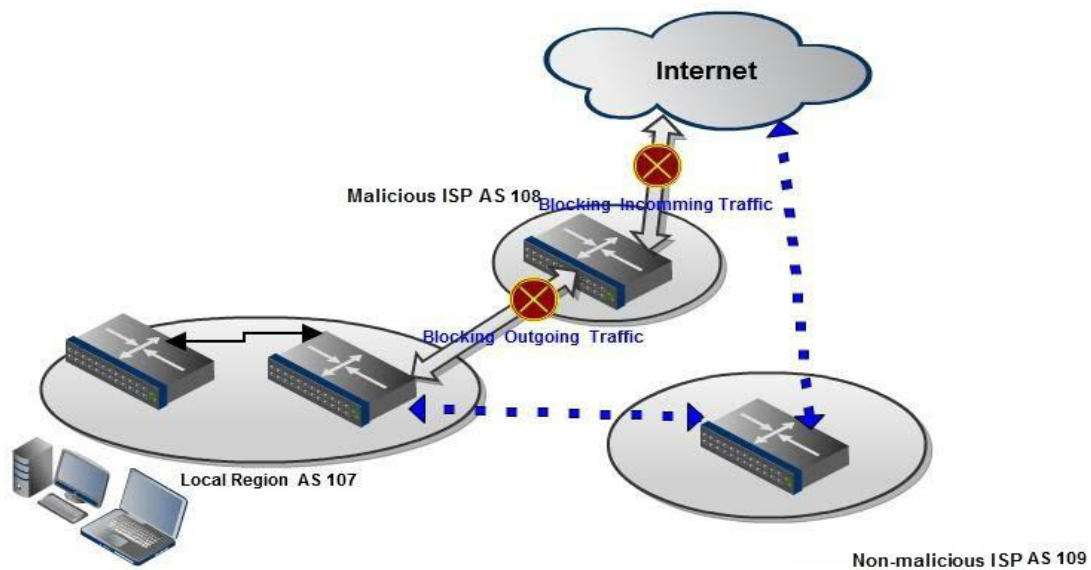


Figure 1. 1 Malicious ISP blocking of the traffic in the region concerned.

malicious ISP AS, good ISP AS, and other ASes. All these parts are running BGP over IPv6. The malicious ISP AS is blocking traffic coming from and going to the local region. Moreover, the malicious ISP AS continues advertising reachability to the blocked region to other ASes.

Hence, the major goal of this study is to evaluate the performance of the BGP-based solutions identified by Alrefai [1] when IPv6 networks are considered, and to provide enhancements whenever possible. The study will be accomplished by conducting OPNET [5] simulations under different scenarios including the use of different traffic loads and network applications. Furthermore, the study will compare the different solutions based on the convergence time and effect on Internet applications.

1.3. Summary of Contributions

- Implementing the problem model and the BGP tuning methods that were proposed by Alrefai [1] using OPNET for IPv6 networks.
- Evaluating the BGP-Based solutions for different application types, different background traffic load, and different Internet delay times.
- Comparing results obtained by Alrefai [1] for IPv4 with our IPv6 results.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

2.1. Background

The vision of the Internet having the users being connected anytime and anywhere is becoming more and more a reality. One of the critical enabling technologies for workstations and servers for global connectivity is the emergence of the Internet, which interconnects different ASes. Users can access the Internet using DSL, cable, wireless, dial-up lines, or any type of Internet access services provided by a local ISP. Local ISPs are categorized as tier-3 ISPs in the Internet structure.

The two types of relationships between ISPs are Transit and Peering. Transit interconnection is a provider-customer relationship. It simply exists when an ISP sells dedicated access to its customer ISPs via private leased-line circuits. The customer ISPs pay for the Internet access in this type of interconnection.

Peering, on the other hand, refers to an interconnection between two ISPs to exchange traffic for the mutual benefit of both parties. Each ISP provides the other ISP with access to its networks and customers' networks. This interconnection does not involve payments for the access service, and hence, it is sometimes called "settlement-free peering" to reflect the fact of cost-free interconnection. There are two types of peering, depending on the physical connections that are used: private peering, where a point-to-point link is used to physically connect the two ISPs, and public peering, where

multiple ISPs are interconnected at an Internet Exchange Point (IXP) using a shared switch fabric [6].

2.2. Internet Protocol 6 (IPv6)

The Internet was built upon IPv4 protocol, which was widely deployed and provided unique global computer addressing and connectivity between computers. However, due to the extended dependency and the high growth of the Internet services, IPv4 suffered from address exhaustion, routing problems [7] and security issues. IPv6 is a new version of the Internet protocol that was designed by the Internet Engineering Task Force (IETF) [8]. IPv6 was mainly proposed to increase the number of bits used in the IP addresses from 32 bits to 128 bits [4]. The following subsection further explains the additional IPv6 features and benefits.

2.2.1. IPv6 Features and Benefits

As IPv6 is the successor of IPv4, it inherits the existing features of IPv4 and provides new services and capabilities. The following is a description of the features and the benefits of IPv6 [8].

Increased Address Space: IPv6 increases the address size from 32 bits to 128 bits. Extending the address space to 128 bits offers the following two additional benefits:

- 1. Better Applications Functionality:*** Since IPv4 suffers from address exhaustion, there have been different solutions proposed to solve this problem such as the use of Network Address Translation (NAT). However, these solutions created

additional problems such as server reachability problems. Accordingly, IPv6 provides a unique IP address to each device which results in simplifying the operation of peer-to-peer applications and networking.

2. ***Enhanced Transparency:*** Each end system will be assigned a unique address; no need for address translation for IPv6, which enhances the transparency.

Streamlined Packet Format: The IPv6 header was designed to reduce the common case processing cost of packet handling, and to keep the bandwidth cost of the IPv6 header as low as possible.

Auto-configuration: For auto-configuration, IPv4 uses DHCP, which is called stateful auto-configuration. IPv6 supports both stateful and stateless auto-configuration. In a stateless auto-configuration, a DHCP server is not required to obtain addresses, and instead it uses router advertisements to create a unique address. Thus, this mechanism offers a “plug-and-play” environment that simplifies address management, and administration configuration and reconfiguration.

Scalability of Multicast: Multicast is the ability to send a single packet to multiple nodes in the network. IPv4 supports multicast by using multicast addresses, but IPv6 provides a much larger pool of multicast addresses with multiple scoping options. IPv6 multicast provides several communication ways with groups, routers or hosts.

Improved Security: IPv4 suffers from different types of security issues such as denial of service, repudiation, sniffing attack and others. For this reason, IPv6 was designed with built-in IPsec protocol that provides much better security enhancement. IPv6 includes the definition of extensions, which provide support for authentication,

data integrity and confidentiality. This is included as a basic element of IPv6 and will be included in all implementations.

Better Quality-of-Service: Traffic handling and identifying have been improved in IPv6 by using new fields that were added to the IPv6 header. For example, the 24-bit Flow Label field in the IPv6 header is a bit sequence that identifies a stream of packets sent from a particular source to a particular destination for which the source desires special handling by the intervening routers. From a networking point of view, the quality of service (QoS) refers to data loss, latency or jitter, and bandwidth. In order to implement QoS marking, IPv6 provides an 8-bit traffic-class field.

Speed: IPv6 will have reduced end-to-end delay when compared with IPv4 due to many reasons. The IPv6 design includes an end-to-end fragmentation that reduces the router's load of handling fragmented packets. By reducing the work required by routers to split and identify data, the overall end-to-end delay is reduced and the workload along the transport path goes down. Moreover, the header of IPv6 has been designed in a way to speed-up the routing process. In addition, IPv6 eliminated the need for integrity-checking of packets during transit, leaving this to higher layer such as Transmission Control Protocol (TCP). As such, an IPv6 router will be able to forward the data faster than in the case of an IPv4 router.

2.3. Border Gateway Protocol (BGP)

Routing in the Internet is categorized into two parts: the internal fine-grained portions that are managed by an Internal Gateway Protocol (IGP), and the interactions between ASes via an External Gateway Protocol (EGP). IGP protocols learn about routes to networks that are internal to the AS, hence the name Interior. Some examples of IGP protocols include Routing Information Protocol (RIP) and Open Shortest Path First Protocol (OSPF). On the other hand, EGP protocols are used for routing between networks, especially on the Internet backbone itself, linking the different ASes together. BGP is the most common EGP in use on the Internet.

2.3.1. BGP Attributes

The following is a list defining and describing important BGP attributes that are used in the BGP path selection process [5]:

Weight: A Cisco-defined attribute that is local to a router. This attribute is NOT advertised to any BGP neighbor. A path with a high weight value is preferred over a path with a low weight value.

Local-preference: Used to influence outbound path selection. If there are multiple exit points out of a BGP AS, then a path with the highest local-preference value will always be the preferred path out. Unlike the weight attribute, the local-preference attribute is propagated throughout the local AS.

Multi-exit discriminator (MED): When a path includes multiple exit or entry points to an AS, this value may be used as a metric to discriminate between them. A path with a lower MED value is preferred over a path with a higher MED value.

AS-Path: This is a list of autonomous system numbers that describes the sequence of ASes through which this route description has passed. This is a critically important attribute since it contains the actual path of autonomous systems to the network. It is used to calculate routes and to detect routing loops.

eBGPmultihop: Multi-hop is used to allow two routers that do not share a direct physical connection to establish a BGP peering session. When a BGP router exchanges routes with another BGP router the BGP peering occurs. There are two types of peering sessions. First, an external BGP (eBGP) is used to establish a connection between two non-directly connected external peers such that the connected peers appear to be neighbors of each other. Second, an internal BGP (iBGP) peering is used inside an autonomous system. The multihop is used only for eBGP and not for iBGP.

Origin: A mandatory attribute that defines the origin of the path information. The origin attribute can assume one of three values as explained in the following table:

ORIGIN Code	ORIGIN Code Name	Description
0	IGP	<p>The route originated on a BGP speaking router.</p> <p>The IGP ORIGIN type is the most preferred ORIGIN for a route during the path selection process and is selected before the EGP or Incomplete ORIGIN types.</p>
1	EGP	<p>The route originated from an EGP (not E-BGP) session. The EGP ORIGIN type is more preferred than the Incomplete ORIGIN type.</p>
2	Incomplete	<p>The route originated from a routing process other than BGP, and entered BGP by means of manual redistribution, such as redistribution from an IGP protocol, static route, or connected route. The Incomplete ORIGIN type is not preferred over IGP or EGP.</p>

Table 1.1 BGP ORIGIN Codes

2.3.2. BGP Path selection Procedure

Routing protocols are responsible for selecting a path between two communicating nodes. There are multiple protocols to route traffic such as BGP and OSPF. These routing protocols use different procedures to select the best path to route traffic from source to destination. As explained in [5], BGP uses a best path algorithm to select the best path to route traffic. BGP selects a best path by choosing the highest weight value of all available paths. If the weight values are the same then the Local-Preference value is compared and the path with the highest value is selected. If the Local-Preference value is the same, the BGP selects the route with the lowest ORIGIN value. If all routes have the same ORIGIN value, then BGP selects the shortest AS-Path length. If the AS-Path length is the same for all paths, then the BGP selection procedure selects the path with the lowest MED value. In the case that all paths have the same MED value, the IBGP path is selected over EBGP. If the paths are the same, BGP selects the route with the lowest IGP cost which is associated with the nearest neighbor. If they are the same, BGP selects the route received from the peer with the lowest BGP router ID.

2.3.3. BGP Threats and Attacks

There are many studies that have been conducted to investigate the Internet resiliency against deliberate and non-deliberate threats and weaknesses. In addition, there have been a number of approaches that have been proposed to recover from an outage that may happen due to specific types of the Internet outage. In the following we

present some of the threats and weaknesses facing the Internet resiliency as well as some of the proposed solutions to recover from specific causes of Internet outages.

The Internet resiliency is highly dependent on the robustness of BGP. There are many issues related to BGP's capability to meet the scale of the growth of the Internet, mainly due to security concerns. Barrett et al. [9] discussed some security issues of BGP. First, BGP does not provide an authorization mechanism to ensure the ownership of a specific block of addresses that are being advertised by a particular AS. Second, BGP does not have any mechanism to make sure the advertising router really has reachability to the advertised path.

Nordstrom and Dovrolis [10] showed that BGP is vulnerable to four different threats. First, blocking the traffic for a specific AS or prefix by dropping the traffic that reaches the attacked router. The second threat for BGP can happen by sending fake updates or advertisements to make the network unstable by advertising unreachable or non-existent paths. Supervision is the third threat, where the attacker redirects the traffic to the originally intended destination but only after modifying it. The fourth threat is achieved when the attacker redirects the traffic to a different destination for inspection before resending it to the original destination without any modifications. Such a threat is referred to as redirection. Similarly, Hu et al. [11] provided a list of security weaknesses of BGP. First, the message integrity and message origin authentication mechanisms are not provided by BGP. Second, BGP does not provide a mechanism to verify the legality of the AS-Path or the prefix advertisements from the AS. Third, attributes in BGP messages are passed on without any validity check.

Kim et al. [12] proved that by modifying the Internet infrastructure, a higher Internet resiliency can be achieved. Cohen et al. [13] proved mathematically that the Internet could suffer from a momentous outage because of deliberate attacks that attack a specific AS that is aggregating a huge number of Internet connections. This kind of attack happens due to the complex and non-structured nature of the Internet architecture.

Dolev et al. [14] measured and analyzed the Internet resiliency based on ASes connectivity as a directed graph. Moreover, they have shown that the Internet is highly resistant to non-deliberate attacks, while it is highly affected by deliberate attacks.

In subsection 2.3.4 we provide a brief description for some of the proposed solutions that address the BGP threats. On the other hand, subsection 2.3.5 provides an explanation of a BGP feature that enhances the Internet resiliency.

2.3.4. BGP Security Solutions

Nordstrom and Dovrolis [8] proposed two types of countermeasures for the BGP threats. Filtering is the first approach which requires that ASes first filter all fake advertisements and updates. Secure Border Gateway Protocol (S-BGP) is the second proposed approach to countermeasure the BGP threats, but on the other hand it will create more overheads such as performance overheads and deployment costs.

Jin and Wang [15] proposed another mechanism which countermeasures BGP threats. The proposed mechanism performs better than S-BGP. The mechanism verifies all announced prefixes through a verification system referred to as the Assignment

Track (AT). The AT requires each AS to verify its address assignment, and accordingly the AT certifies that announced prefix by an AS belongs to that AS.

2.3.5. BGP Multihoming

BGP multihoming enables a BGP router to connect a site to two or more ASes to provide redundant connectivity and increase the Internet resiliency. As explained by Liu and Xiao [16], the two main types of multihoming are BGP multihoming and NAT multihoming. BGP multihoming is the ability of stub networks to connect to two or more public network connections to the Internet using BGP. BGP multihoming guarantees the uniqueness of the host IP address. On the other hand, NAT multihoming is based on the use of NATing to map a number of public Internet addresses assigned by different ISPs to internal local network addresses. Savola [17] has shown the functionalities and the restrictions of the IPv6 multihoming such as maintain connection survivability when network outage happens and the multihomed site loses physical connection to one of the ISPs. Moreover, they provided the main steps to achieve new multihoming architecture for IPv6.

CHAPTER 3

IMPLEMENTATION & VALIDATION OF SOLUTIONS

USING BGP TUNING BASED APPROACH

3.1.Introduction

In this chapter, we evaluate the impact of implementing the BGP-Based solutions that were proposed by Alrefai [1] to solve the problem of the Internet access denial problem caused by malicious ISPs. The evaluation considers applying the following methods to an IPv6 network:

1. Local- Preference: This method relies on the fact that a route with a higher local preference is more preferred. The local preference is used to control the outgoing traffic by assigning a higher local preference value in the local region gateway router to the non-malicious ISP. This will make the non-malicious ISP more preferred and will direct the outgoing traffic through the non-malicious ISP.

2. Community: The community method is used to control the incoming traffic by using the BGP community attribute. The BGP community attribute can be set to a specific value and combined with a specific prefix before advertising it to other ASes. When an AS receives the prefix it will check the community value and if it matches the specific community number it triggers the AS to assign the non-malicious ISP a higher local preference value to make the route through the non-malicious ISP more preferred. As a result, the traffic to the local region will go through the non-malicious ISP.

3. More Specific Prefix: The routing tables contain address prefixes to be used for comparing the destination address of an incoming packet with such address prefix address. Since the router uses the longest prefix matching rule, and if the non-malicious ISP routing advertises a longer prefix to the local region, then this will make the non-malicious ISP more preferred to the other routers. Thus, the more specific prefix method controls the incoming traffic through advertising more specific prefixes to the local region by the non-malicious ISP. As a result, the non-malicious ISP becomes more preferred to the other routers when sending the traffic to the local region.

4. Prepending: AS Path Prepending is a common BGP method to influence path selection. Prepending works by adding an AS number to the end of the path one or more times. Adding the AS number one or more times makes the AS-Path longer and less preferred. When prepending the local region AS and advertising it to the malicious ISP, the malicious ISP becomes less preferred to the other routers while the non-malicious ISP becomes more preferred to the other routers in sending the traffic to the local region.

5. AS Path Shortening: BGP prefers the shortest AS path when selecting between different paths. Hence, in the shortening method the non-malicious ISP advertises the local region prefix without the AS number of the local region. Thus, the advertised local region prefix by the non-malicious ISP will be shorter than that advertised by the malicious ISP. As a result, the non-malicious ISP becomes more preferred by the other routers in directing the traffic to the local region.

We will refer to the aforementioned methods as BGP tuning methods. In order to evaluate the impact of implementing the BGP tuning methods on the network to

control outgoing and incoming traffic, OPNET simulations are performed. OPNET is the industry's leading network development software. OPNET provides the ability to design and study communication networks, devices, protocols, and applications. OPNET's object-oriented modeling approach and graphical user interface (GUI) enable relatively easy means of developing models from the actual world network, hardware devices, and protocols. OPNET supports all major network types and technologies, allowing the design and testing of various scenarios with reasonable certainty of the output results. OPNET version 17.5 PL3 supports most features of the IPv6 and BGP-4 protocols. OPNET gives the ability to work with EBGp and IBGP connections. Moreover, the GUI provides easy way to create BGP policies and change the BGP attributes which allows us to simulate the control of the incoming and the outgoing traffic. Prepending, use of Community and Local Preference are supported by OPNET. On the other hand, OPNET does not support changing the configuration of the simulation while it is running. In addition, AS-Path, shortening, more specific prefixes, and malicious blocking by an ISP are not supported in OPNET.

This chapter is organized as follows. The BGP simulation for the baseline configuration is presented first. Then, the basic implementation for controlling incoming and outgoing traffic is shown. Finally, the code changes that have been made to provide real scenarios, to make changes in the middle of the simulation, and to add non-supported solutions such as AS-Path shortening and more specific prefixes as well as the associated simulation are presented.

3.2. General Methodology

The OPNET simulations and evaluation of the IPv6 BGP-based solutions are done by considering different Internet environment scenarios. Figure 3.1 shows the baseline network configuration for our simulation. AS12 is the local region and it represents the region of concern. The local region is a multihomed AS with two ISPs. The primary ISP is AS3 and it is referred to as the malicious ISP. On the other hand, AS4 is the secondary ISP for the local region and it is referred to as the non-malicious ISP. Moreover, AS7 includes the client side that request services from AS12 that hosts application servers that provide FTP, HTTP and VoIP services. The malicious ISP deliberately drops the traffic for the local region while it still advertises the local region prefixes on the Internet. In addition, the malicious ISP continues to exchange keep alive and BGP messages with the local region speaker router. When the local region detects the loss of data exchange with the malicious router then the local region speaker router will attempt to force the outgoing traffic and draw the incoming traffic through the non-malicious ISP.

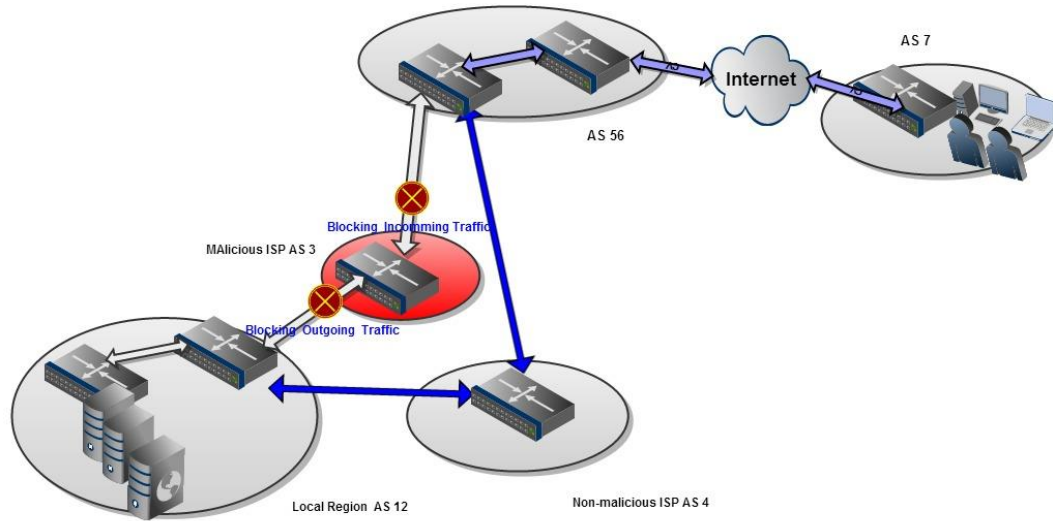


Figure 3. 1 Base Simulation Scenario.

Incoming and outgoing scenarios are tested with the same procedure. The testing procedure consists of different traffic configurations that combine a tested network application with specific traffic load. With each scenario tested, the performance figures of convergence time, throughput, end-to-end delay and packet loss are collected when switching from the malicious ISP to the non-malicious ISP. The results obtained through this simulation for IPv6 are compared against the results obtained for IPv4 and that are reported by Alrefai [1].

3.3. BGP-Based Solutions

The BGP-based solutions to be tested are those used by Alrefai [1]. More specifically, the following are the tested solutions: Local-Preference, AS-Path Shortening, More Specific Prefix, use of Community, and Prepending.

3.4. Baseline Simulation Configuration

In this study, we have to test different types of solutions such as solutions supported directly by OPNET, and solutions provided through code modifications. In each case, we try to control incoming and outgoing IPv6 traffic. The network setup used in this study has the same structure and components as the network setup used by Alrefai [1], and is shown in Figure 3.2.

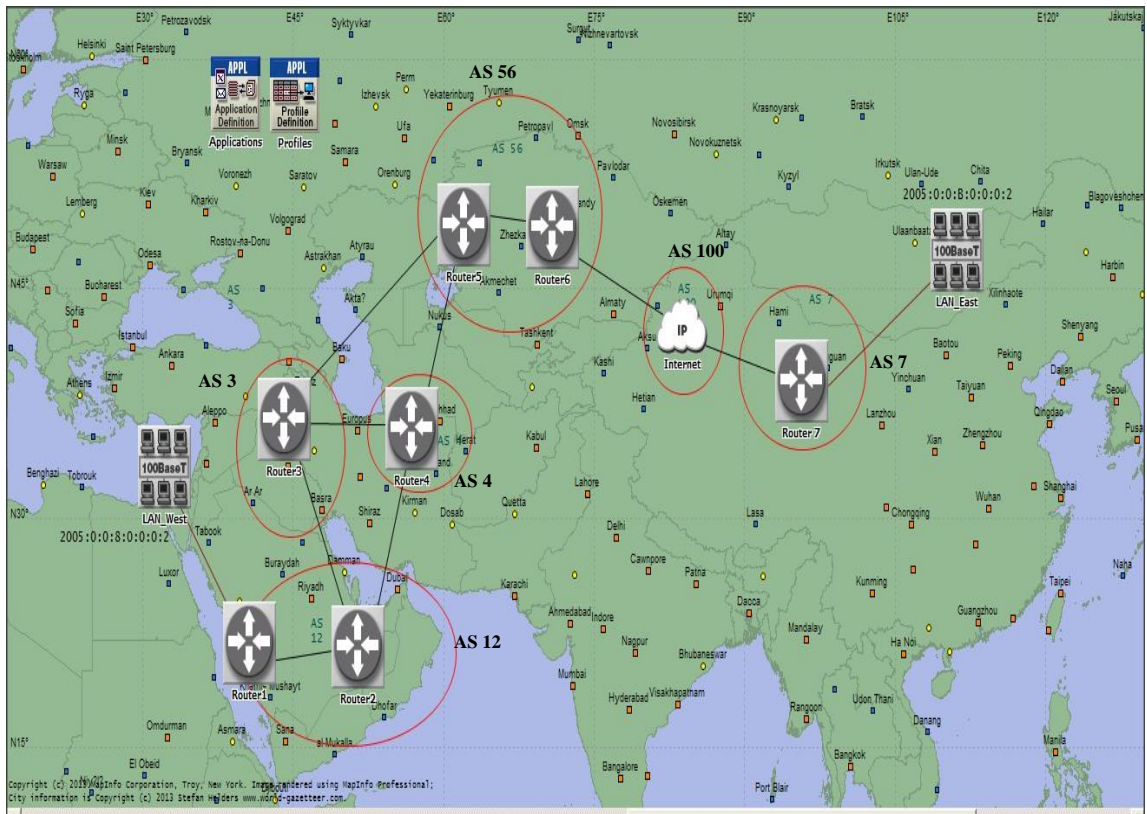


Figure 3. 2 Baseline Network Configuration.

3.5. Devices Used

ethernet4_slip8_gtwy router: is an IP-based gateway router that supports four Ethernet hub interfaces and eight serial line interfaces. The IP packets are routed based

on the destination IP address at any interface. A gateway router supports almost all protocols such as IP, UDP, TCP, BGP, Ethernet, and other protocols.[4]

As shown in Figure 3.2, we have 7 routers, all of them are configured with IPv6 and BGP protocols. IBGP protocol is configured inside each autonomous system routers and EBGP is configured between different autonomous systems.

100BaseT_LAN object models: used to simulate Ethernet LANs running over 100BASE-X, 100BASE-T, and 10BASE-T. Each model simulates the operations of a LAN with a number of end nodes. Moreover, this model can be configured with different configurations such as applications, switching speed and the number of workstations. [4]

In our simulation, we assume that LAN_West that is connected to AS12 communicates with LAN_East that is connected to AS7. LAN_West provides different types of services such as HTTP, FTP, and VoIP to LAN_East.

ip32_cloud node: models an IP cloud and is commonly used to represent the connectivity to the Internet. This model has 32 serial IP interfaces. This node is used to simulate the delay of the Internet by configuring ‘Packet Latency’ attributes and ‘Packet Discard Ratio’ used to specify the percentage of traffic to be discarded. Through this node we can observe the effect of Internet delay on the convergence time of the different solutions. In the base_line scenario, the delay will be 1 ms, whereas for the solutions scenarios we test different delay values.

Bidirectional PPP_DS3 link: a link that has a data rate of 44.736 Mbps and is used to connect nodes that run the IP protocol using ip3_dgram packet format [4]. In our simulation, this type of link is used to connect routers with each other.

Bidirectional 100BaseT link: is a link that works at 100Mbps and has a “Propagation Speed” attributes that can be configured. In our network we use this link to connect LANs to their routers.

3.6. Controlling Traffic Using Supported OPNET features

Using the routing policy in OPNET allows the control of the outgoing traffic from the local ISP and the incoming traffic to the local ISP. Note that the local ISP is multihomed to malicious and non-malicious ISPs. The outgoing traffic is controlled using two methods. First, by using the Local-Preference property which is supported by OPNET. Second, by modifying the AS-Path through prepending to make the routes through a specific route less preferred. OPNET supports the use of community that allows ASes to prefer routes with a certain community number. In this section, we will use the supported configurations by OPNET to control the incoming and outgoing traffic. No malicious router will be configured for this section.

3.6.1. Baseline Simulation

In our simulation, we assume that LAN_West that is connected to AS12 responds to HTTP, FTP, and VoIP requests from LAN_East that is connected to AS7. AS12 consists of Router1 and Router2, and represents the local ISP, LAN_West is the network that the malicious router, Router3, is targeting for blocking. Router2 is a speaker router for the local region that is connected to Router3 and Router4 that belong

to the malicious ISP and non-malicious ISP, respectively. Since Router3 is the malicious ISP, it will be configured to block incoming/outgoing traffic to the local ISP.

The main traffic we are looking for is the IPv6 traffic between Router2 and Router3 and between Router2 and Router4 in both directions as shown in Figure 3.3.

In Figure 3.3, time, measured in seconds, is shown in the x-axis, while the y-axis represents the throughput in packets/second. As shown in Figure 3.3, both data traffic and BGP traffic are exchanged between Router2 and Router3, whereas only BGP traffic is exchanged between Router2 and Router4.

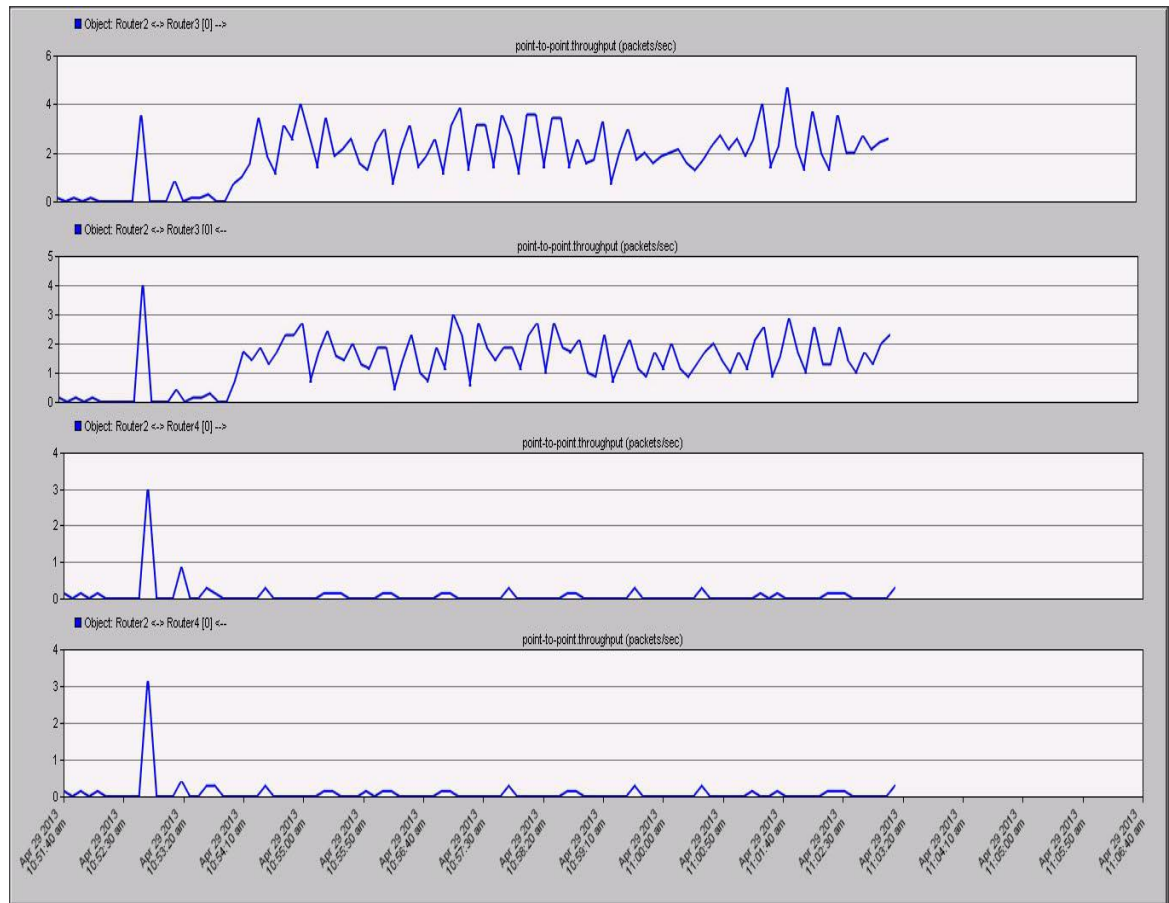


Figure 3. 3 Incoming and Outgoing Traffic in Baseline Simulation.

Figure 3.4 shows the IP forwarding table for Router2. The IP address of LAN_East is 2005:0:0:B:0:0:0:1 so it belongs to the prefix 2005:0:0:B/64. It can be seen from Figure 3.4 that Router3 is the 'Next Hop Node' for the LAN_East prefix since Router3 is the primary ISP for the local region.

	Destination	Source Protocol	Route Preference	Metric	Next Hop Address	Next Hop Node	Outgoing Interface
1	2005:0:0:1:0:0:0:0/64	Direct	0	0	2005:0:0:1:0:0:0:1	Router2	IF11
2	2005:0:0:1:0:0:0:1/128	Local	0	0	2005:0:0:1:0:0:0:1	Router2	IF11
3	2005:0:0:2:0:0:0:0/64	BGP	20	0	2005:0:0:1:0:0:0:2	Router3	IF11
4	2005:0:0:3:0:0:0:0/64	Direct	0	0	2005:0:0:3:0:0:0:1	Router2	IF10
5	2005:0:0:3:0:0:0:1/128	Local	0	0	2005:0:0:3:0:0:0:1	Router2	IF10
6	2005:0:0:5:0:0:0:0/64	Direct	0	0	2005:0:0:5:0:0:0:1	Router2	IF4
7	2005:0:0:5:0:0:0:1/128	Local	0	0	2005:0:0:5:0:0:0:1	Router2	IF4
8	2005:0:0:6:0:0:0:0/64	BGP	20	0	2005:0:0:1:0:0:0:2	Router3	IF11
9	2005:0:0:7:0:0:0:0/64	BGP	20	0	2005:0:0:1:0:0:0:2	Router3	IF11
10	2005:0:0:8:0:0:0:0/64	RIPng	120	11	2005:0:0:3:0:0:0:2	Router1	IF10
11	2005:0:0:9:0:0:0:0/64	BGP	20	0	2005:0:0:1:0:0:0:2	Router3	IF11
12	2005:0:0:8:0:0:0:0/64	BGP	20	0	2005:0:0:1:0:0:0:2	Router3	IF11
13	2005:0:0:C:0:0:0:0/64	Direct	0	0	2005:0:0:C:0:0:0:1	Router2	LB0
14	2005:0:0:C:0:0:0:1/128	Local	0	0	2005:0:0:C:0:0:0:1	Router2	LB0
15	2005:0:0:D:0:0:0:0/64	BGP	20	10	2005:0:0:1:0:0:0:2	Router3	IF11
16	2005:0:0:E:0:0:0:0/64	BGP	20	0	2005:0:0:1:0:0:0:2	Router3	IF11
17	2005:0:0:F:0:0:0:0/64	RIPng	120	11	2005:0:0:3:0:0:0:2	Router1	IF10
18	2005:0:0:10:0:0:0:0/64	BGP	20	10	2005:0:0:5:0:0:0:2	Router4	IF4
19	2005:0:0:18:0:0:0:0/64	BGP	20	0	2005:0:0:1:0:0:0:2	Router3	IF11
20							

Figure 3. 4 IP Forwarding Table for Router2.

Similarly, Figure 3.5 shows the IP forwarding table of Router5. The IP address of LAN_West is 2005:0:0:8:0:0:0:2, so it belongs to the prefix 2005:0:0:8/64. It can be seen from Figure 3.5 that Router3 is the ‘Next Hop Node’ for the LAN_West prefix.

	Destination	Source Protocol	Route Preference	Metric	Next Hop Address	Next Hop Node	Outgoing Interface
1	2005:0:0:1:0:0:0:0/64	BGP	20	0	2005:0:0:2:0:0:0:1	Router3	IF10
2	2005:0:0:2:0:0:0:0/64	Direct	0	0	2005:0:0:2:0:0:0:2	Router5	IF10
3	2005:0:0:2:0:0:0:2/128	Local	0	0	2005:0:0:2:0:0:0:2	Router5	IF10
4	2005:0:0:3:0:0:0:0/64	BGP	20	0	2005:0:0:2:0:0:0:1	Router3	IF10
5	2005:0:0:5:0:0:0:0/64	BGP	20	0	2005:0:0:2:0:0:0:1	Router3	IF10
6	2005:0:0:6:0:0:0:0/64	Direct	0	0	2005:0:0:6:0:0:0:1	Router5	IF11
7	2005:0:0:6:0:0:0:1/128	Local	0	0	2005:0:0:6:0:0:0:1	Router5	IF11
8	2005:0:0:7:0:0:0:0/64	Direct	0	0	2005:0:0:7:0:0:0:1	Router5	IF4
9	2005:0:0:7:0:0:0:1/128	Local	0	0	2005:0:0:7:0:0:0:1	Router5	IF4
10	2005:0:0:8:0:0:0:0/64	BGP	20	0	2005:0:0:2:0:0:0:1	Router3	IF10
11	2005:0:0:9:0:0:0:0/64	RIPng	120	11	2005:0:0:7:0:0:0:2	Router6	IF4
12	2005:0:0:8:0:0:0:0/64	IBGP	200	0	2005:0:0:18:0:0:0:1	Router 7	Unresolved
13	2005:0:0:C:0:0:0:0/64	BGP	20	0	2005:0:0:2:0:0:0:1	Router3	IF10
14	2005:0:0:D:0:0:0:0/64	BGP	20	10	2005:0:0:2:0:0:0:1	Router3	IF10
15	2005:0:0:E:0:0:0:0/64	Direct	0	0	2005:0:0:E:0:0:0:1	Router5	LB0
16	2005:0:0:E:0:0:0:1/128	Local	0	0	2005:0:0:E:0:0:0:1	Router5	LB0
17	2005:0:0:F:0:0:0:0/64	BGP	20	0	2005:0:0:2:0:0:0:1	Router3	IF10
18	2005:0:0:10:0:0:0:0/64	BGP	20	10	2005:0:0:6:0:0:0:2	Router4	IF11
19	2005:0:0:18:0:0:0:0/64	RIPng	120	11	2005:0:0:7:0:0:0:2	Router6	IF4
20							

Figure 3. 5 IP forwarding table of Router5.

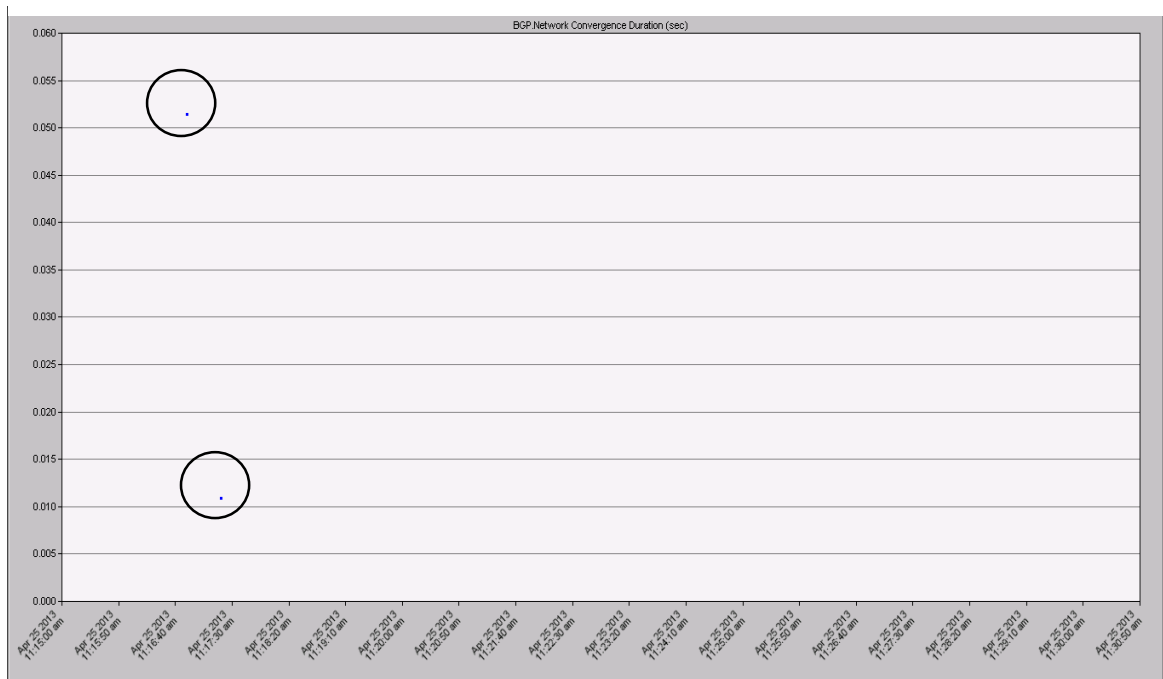


Figure 3. 6 Convergence activity and duration of baseline Simulation

Figure 3.6 shows the BGP convergence activity and duration of the baseline simulation scenario. The Figure shows that there are two convergence activities represented as two points in the figure. The first convergence activity happens because of the start of the BGP which takes about 0.052 seconds. The second activity takes about 0.010 seconds. The routing updates are sent as soon as the BGP routing table changes. There are 30 seconds between the two activities and this delay happened because of minimum route advertisement interval (MRAI). The MRAI round is the minimum time interval between sending two consecutive update messages for the same destination. The BGP convergence time is affected by the duration of MRAI and the implementation of MRAI timers. The default MRAI value (30 s) is used in the majority of today's routers and in our simulation.

3.6.2. Outgoing Traffic Control Simulation

To control the outgoing traffic, a high Local-Preference value for the preferred AS was set in Router2 of the simulation. The Local-Preference is one of the BGP attributes that plays a major role in the BGP selection process. Moreover, there is a default Local-Preference value for each neighbor. By applying a policy that assigns higher Local-Preference value for the non-malicious ISP, then all routes learned from the non-malicious ISP will have higher local preference and will be selected as best route. In our simulation we configure a policy in Router2 that gives higher Local-Preference for Router4 to be selected as the best route. Figure 3.7 shows the outgoing traffic from Router 2 to Router3 and from Router2 to Router4.

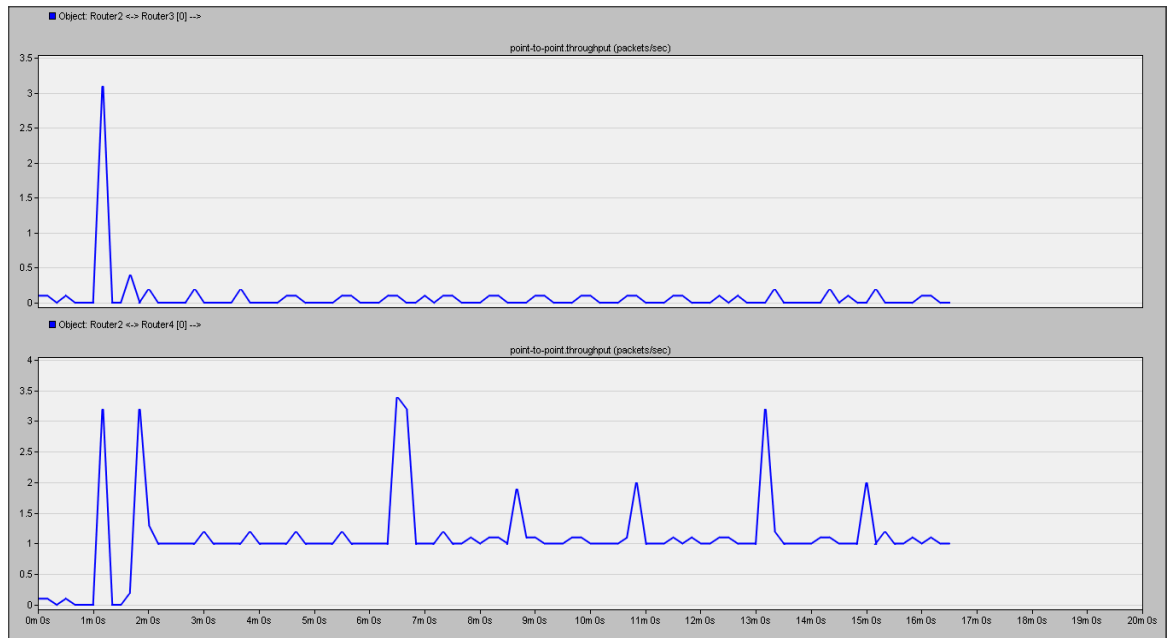


Figure 3.7 IP Outgoing traffic

As Router2 has a higher Local-Preference value set for Router4, we can observe from Figure 3.7 that Router2 traffic is directed to pass through Router4. To check this further, the BGP routing table of Router2 in Figure 3.8 shows that Router4 is the ‘Next Hop Node’ for the destination LAN_East prefix.

	Destination	Source Protocol	Next Hop Address	Next Hop Node	Outgoing Interface	MED	Local Preference	Weight	AS Path	Origin
1	2005:0:0:1:0:0:0:0/64	IBGP	2005:0:0:F:0:0:0:1	Router1	IF10	10	100	0		Incomplete
2	2005:0:0:2:0:0:0:0/64	EBGP	2005:0:0:5:0:0:0:2	Router4	IF4	0	150	0	4 56	Incomplete
3	2005:0:0:3:0:0:0:0/64	Direct	0:0:0:0:0:0:0:0	Unresolved	Unreachable	10	100	32768		Incomplete
4	2005:0:0:5:0:0:0:0/64	IBGP	2005:0:0:F:0:0:0:1	Router1	IF10	10	100	0		Incomplete
5	2005:0:0:6:0:0:0:0/64	EBGP	2005:0:0:5:0:0:0:2	Router4	IF4	0	150	0	4 56	Incomplete
6	2005:0:0:7:0:0:0:0/64	EBGP	2005:0:0:5:0:0:0:2	Router4	IF4	0	150	0	4 56	Incomplete
7	2005:0:0:8:0:0:0:0/64	RIPng	0:0:0:0:0:0:0:0	Unresolved	Unreachable	10	100	32768		Incomplete
8	2005:0:0:9:0:0:0:0/64	EBGP	2005:0:0:5:0:0:0:2	Router4	IF4	0	150	0	4 56	Incomplete
9	2005:0:0:8:0:0:0:0/64	EBGP	2005:0:0:5:0:0:0:2	Router4	IF4	0	150	0	4 56 100 7	Incomplete
10	2005:0:0:C:0:0:0:0/64	IBGP	2005:0:0:F:0:0:0:1	Router1	IF10	10	100	0		Incomplete
11	2005:0:0:D:0:0:0:0/64	EBGP	2005:0:0:5:0:0:0:2	Router4	IF4	0	150	0	4 3	Incomplete
12	2005:0:0:E:0:0:0:0/64	EBGP	2005:0:0:5:0:0:0:2	Router4	IF4	0	150	0	4 56	Incomplete
13	2005:0:0:F:0:0:0:0/64	RIPng	0:0:0:0:0:0:0:0	Unresolved	Unreachable	10	100	32768		Incomplete
14	2005:0:0:10:0:0:0:0/64	EBGP	2005:0:0:5:0:0:0:2	Router4	IF4	10	150	0	4	Incomplete
15	2005:0:0:18:0:0:0:0/64	EBGP	2005:0:0:5:0:0:0:2	Router4	IF4	0	150	0	4 56	Incomplete
16										

Figure 3.8 Forwarding Table of Router 2 After Applying Local-Preference Policy. from Router2.

Figure 3.9 shows that the convergence activity and duration when a higher Local- Preference value is set in the simulation is similar to that found for the baseline experiment.

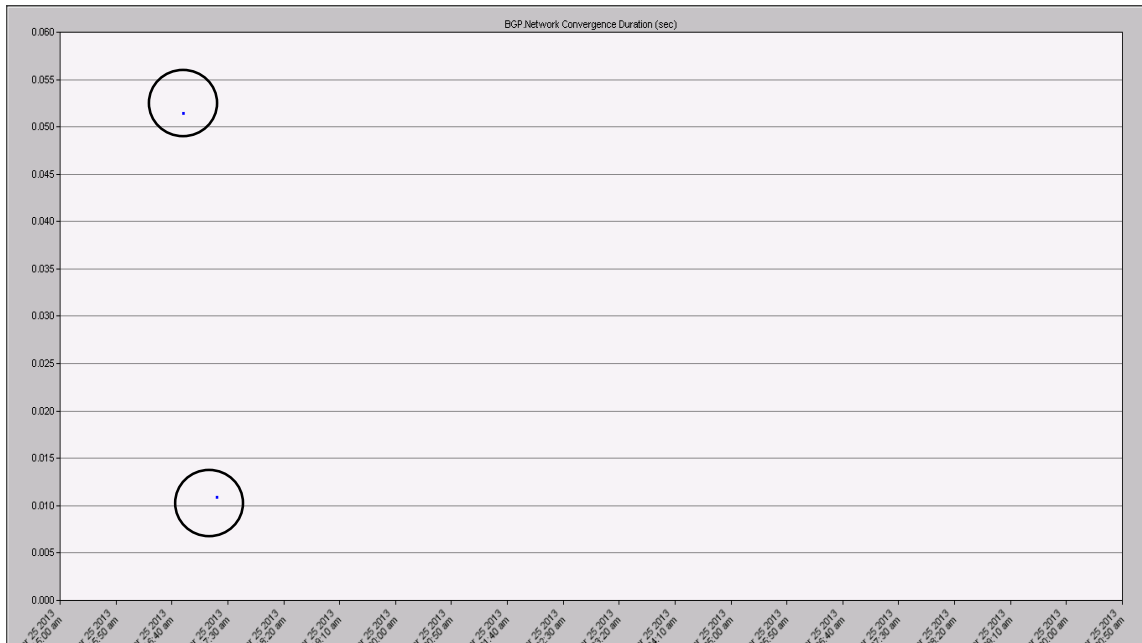


Figure 3. 9 Convergence activity of Local-Preference Policy Scenario.

3.6.3. Incoming Traffic Control Simulations

In this section we present how to control the incoming traffic using prepending and community which are supported by OPNET.

3.6.3.1. Use of Prepending

Prepending is the action of adding your own AS number to the end of the path one or more times, and announcing the prepended path to external BGP peers. Prepending an AS path makes a shorter AS path look longer and therefore less preferable to BGP. The neighbor that receives prepended update messages will also announce the long AS-path and this makes it less preferable for the incoming traffic. We prepended the advertisement that was sent to the malicious ISP (Router3) and that

makes it less preferable while sending normal advertisements to the non-malicious ISP (Router4). Figure 3.10 shows that traffic is incoming to Router2 from Router4 since the routing policy is configured to prepend the AS-Path with AS12 when sending the advertisements to the malicious ISP (Router3).

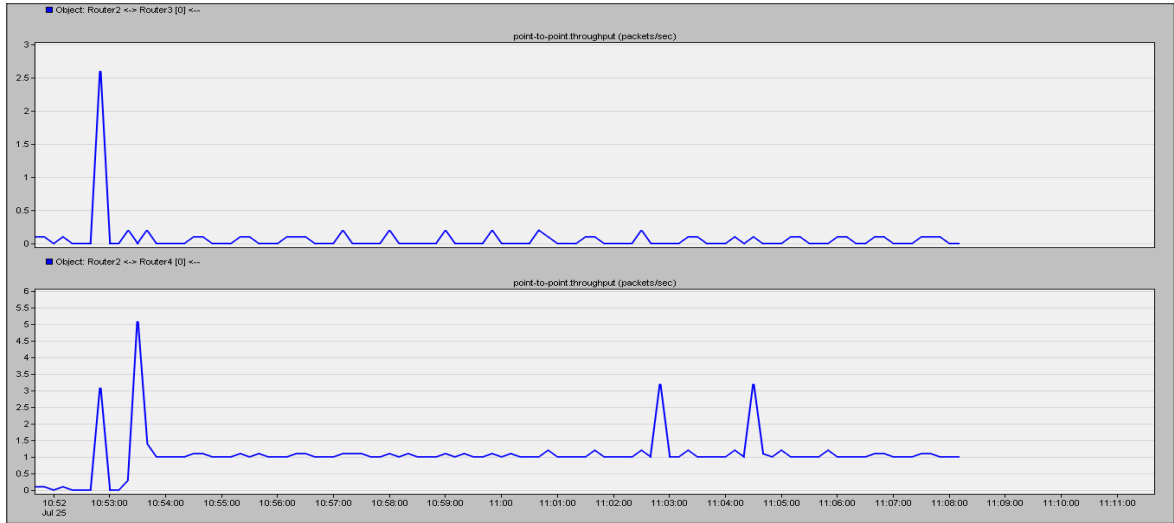


Figure 3. 10 Incoming traffic to Router2 of prepending scenario.

As shown in Figure 3.11, the prepending is achieved by sending BGP update messages from Router2 to Router3 that has the AS 12 prepended. As a result of the BGP update message, Router3 will be forced to reconfigure its BGP routing table to increase the route length between Router 2 and Router3 to 2. On the other hand, the path length is kept at 1 in the BGP routing table of Router5. This makes Router4 more preferred for Router5 over Router3 as shown in Figure 3.11.

	Destination	Source Protocol	Next Hop Address	Next Hop Node	Outgoing Interface	MED	Local Preference	Weight	AS Path	Origin
1	2005:0:0:1:0:0:0:0/64	EBGP	2005:0:0:1:0:0:0:1	Router2	IF10	0	100	0	12 12	Incomplete
2	2005:0:0:2:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:2	Router5	IF4	0	100	0	56	Incomplete
3	2005:0:0:3:0:0:0:0/64	EBGP	2005:0:0:1:0:0:0:1	Router2	IF10	10	100	0	12 12	Incomplete
4	2005:0:0:5:0:0:0:0/64	EBGP	2005:0:0:1:0:0:0:1	Router2	IF10	0	100	0	12 12	Incomplete
5	2005:0:0:6:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:2	Router5	IF4	0	100	0	56	Incomplete
6	2005:0:0:7:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:2	Router5	IF4	10	100	0	56	Incomplete
7	2005:0:0:8:0:0:0:0/64	EBGP	2005:0:0:1:0:0:0:1	Router2	IF10	10	100	0	12 12	Incomplete
8	2005:0:0:9:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:2	Router5	IF4	10	100	0	56	Incomplete
9	2005:0:0:8:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:2	Router5	IF4	0	100	0	56 100 7	Incomplete
10	2005:0:0:C:0:0:0:0/64	EBGP	2005:0:0:1:0:0:0:1	Router2	IF10	0	100	0	12 12	Incomplete
11	2005:0:0:D:0:0:0:0/64	Direct	0:0:0:0:0:0:0:0	Unresolved	Unreachable	10	100	32768		Incomplete
12	2005:0:0:E:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:2	Router5	IF4	0	100	0	56	Incomplete
13	2005:0:0:F:0:0:0:0/64	EBGP	2005:0:0:1:0:0:0:1	Router2	IF10	10	100	0	12 12	Incomplete
14	2005:0:0:10:0:0:0:0/64	EBGP	2005:0:0:4:0:0:0:2	Router4	IF11	10	100	0	4	Incomplete
15	2005:0:0:18:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:2	Router5	IF4	10	100	0	56	Incomplete
16										

Figure 3. 11 BGP routing table of Router3 in Prepend scenario.

As shown in Figure 3.12, Router5 selected Router4 as the “Next Hop Node” to the prefix 2005:0:0:8:0:0:0:0 because it has a shorter AS-Path [4 12] through Router4 than through Router3 which has the AS-Path [3 12 12].

Performance.Routing Table - BGP (IPv6) at 700 seconds for Router5									
	Destination	Source Protocol	Next Hop Address	Next Hop Node	Outgoing Interface	Local Preference	Weight	AS Path	Origin
1	2005:0:0:1:0:0:0:0/64	EBGP	2005:0:0:6:0:0:0:2	Router4	IF11	100	0	4 12	Incomplete
2	2005:0:0:2:0:0:0:0/64	IBGP	2005:0:0:18:0:0:0:1	Router 7	IF4	100	0		Incomplete
3	2005:0:0:3:0:0:0:0/64	EBGP	2005:0:0:6:0:0:0:2	Router4	IF11	100	0	4 12	Incomplete
4	2005:0:0:5:0:0:0:0/64	EBGP	2005:0:0:6:0:0:0:2	Router4	IF11	100	0	4 12	Incomplete
5	2005:0:0:6:0:0:0:0/64	IBGP	2005:0:0:18:0:0:0:1	Router 7	IF4	100	0		Incomplete
6	2005:0:0:7:0:0:0:0/64	Direct	0:0:0:0:0:0:0:0	Unresolved	Unreachable	100	32768		Incomplete
7	2005:0:0:8:0:0:0:0/64	EBGP	2005:0:0:6:0:0:0:2	Router4	IF11	100	0	4 12	Incomplete
8	2005:0:0:9:0:0:0:0/64	RIPng	0:0:0:0:0:0:0:0	Unresolved	Unreachable	100	32768		Incomplete
9	2005:0:0:8:0:0:0:0/64	IBGP	2005:0:0:18:0:0:0:1	Router 7	IF4	100	0	100 7	Incomplete
10	2005:0:0:C:0:0:0:0/64	EBGP	2005:0:0:6:0:0:0:2	Router4	IF11	100	0	4 12	Incomplete
11	2005:0:0:D:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:1	Router3	IF10	100	0	3	Incomplete
12	2005:0:0:E:0:0:0:0/64	IBGP	2005:0:0:18:0:0:0:1	Router 7	IF4	100	0		Incomplete
13	2005:0:0:F:0:0:0:0/64	EBGP	2005:0:0:6:0:0:0:2	Router4	IF11	100	0	4 12	Incomplete
14	2005:0:0:10:0:0:0:0/64	EBGP	2005:0:0:6:0:0:0:2	Router4	IF11	100	0	4	Incomplete
15	2005:0:0:18:0:0:0:0/64	RIPng	0:0:0:0:0:0:0:0	Unresolved	Unreachable	100	32768		Incomplete
16									

Figure 3. 12 BGP routing table of Router5 in Prepend scenario

Figure 3.13 shows the convergence activity for the network. The behavior is similar to that of the baseline experiment.

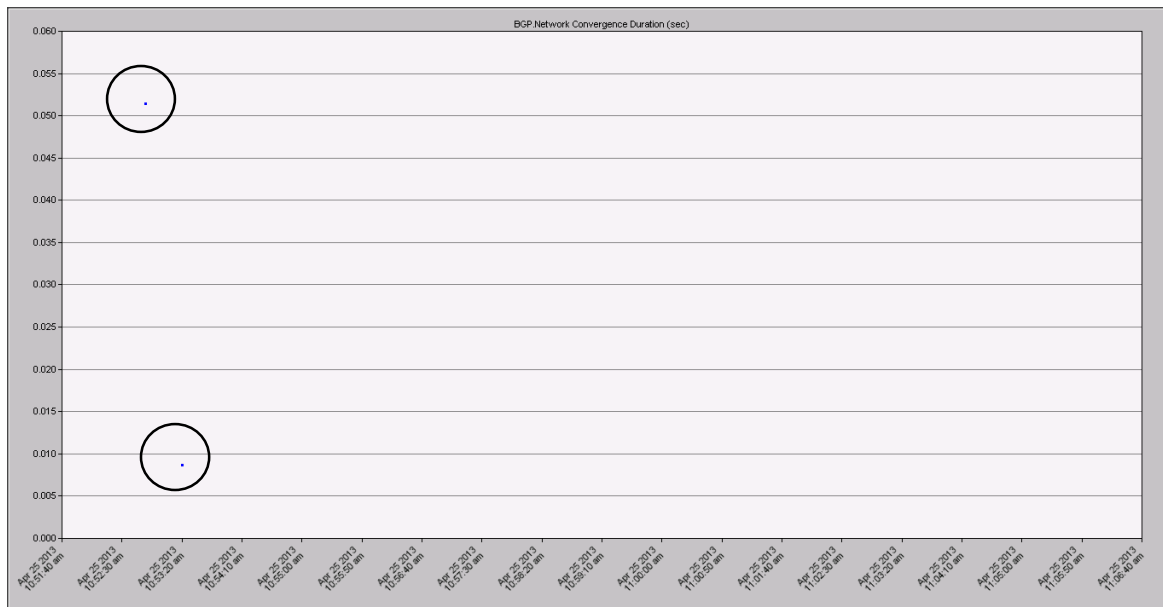


Figure 3. 13 Convergence activity of Prepending policy Scenario.

3.6.3.2. Use of Community

The second approach of controlling incoming traffic using OPNET supported properties is use of community. The community is a BGP numeric attribute that can be assigned to a specific prefix and advertised to other neighbors. When the neighbor receives the prefix it will examine the community value and take proper action whether it is filtering or modifying other attributes. The use of community attributes requires agreement between ASes that will use it.

In our experiment we need to control the traffic that comes from Router5 to go through Router4. The agreement between our local ISP Router2 and Router5 is to assign a higher local preference for the route that announces an advertisement with a specific community number. Accordingly, Router2 is configured using a route map to assign all advertised routes from Router2 with community number 12:144 (12 the AS number and 150 is our community number) and then applies this route map for every route advertised to Router4. On the other hand, Router5 will examine all the received routes and when any route comes with the same community number it gives that route a higher local preference. As shown in Figure 3.14 the throughput of incoming traffic to Router2 from Router3 and from Router4.

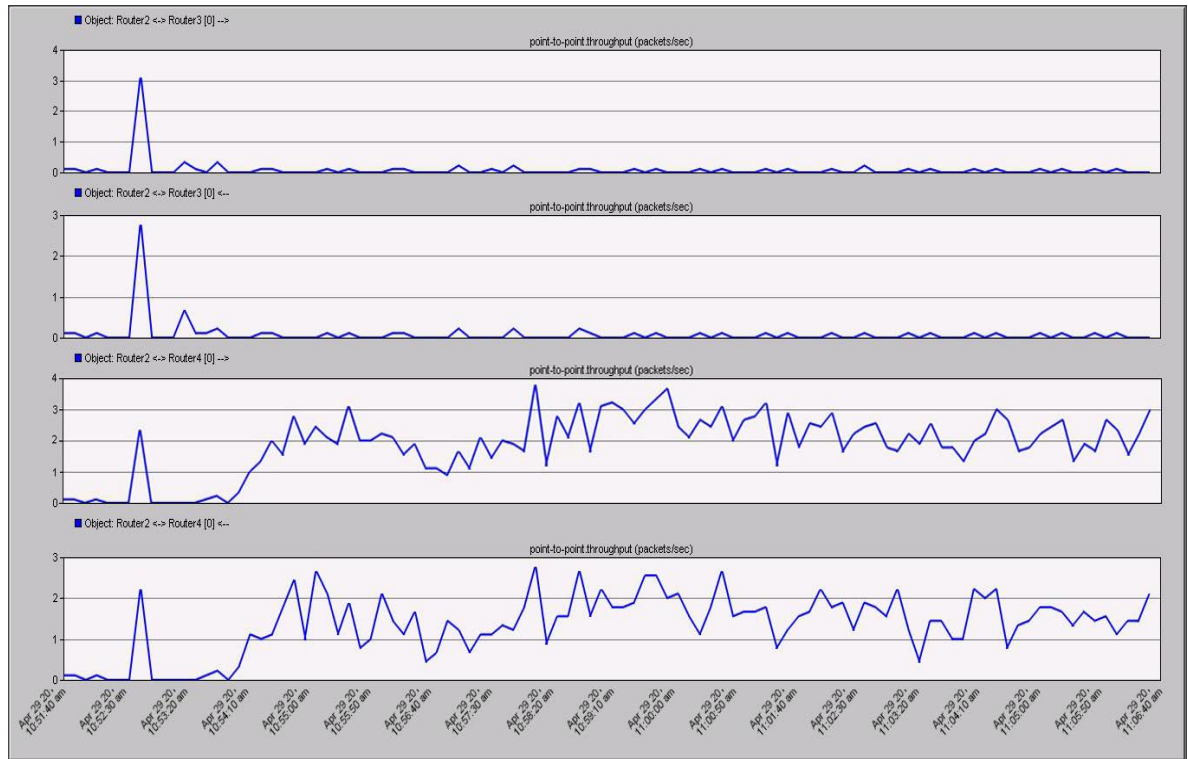


Figure 3. 14 Incoming Traffic to Router2 Using Community.

After applying the community approach, Figure 3.14 shows more traffic flowing between Router2 and Router4 than between Router2 and Router3. Thus, Router2 prefers Router4 over Router3 when exchanging traffic with LAN_East. To see how the community approach works, Figure 3.15 shows the BGP routing table of Router5.

Performance.Routing Table - BGP (IPv6) at 2000 seconds for Router5											
	Destination	Source Protocol	Next Hop Address	Next Hop Node	Outgoing Interface	MED	Local Preference	Weight	AS Path	Origin	Community
1	2005:0:0:1:0:0:0:0/64	EBGP	2005:0:0:6:0:0:0:2	Router4	IF11	0	150	0	4 12	Incomplete	[12:144]
2	2005:0:0:2:0:0:0:0/64	IBGP	2005:0:0:18:0:0:0:1	Router7	IF4	10	100	0		Incomplete	
3	2005:0:0:3:0:0:0:0/64	EBGP	2005:0:0:6:0:0:0:2	Router4	IF11	0	150	0	4 12	Incomplete	[12:144]
4	2005:0:0:5:0:0:0:0/64	EBGP	2005:0:0:6:0:0:0:2	Router4	IF11	0	150	0	4 12	Incomplete	[12:144]
5	2005:0:0:6:0:0:0:0/64	IBGP	2005:0:0:18:0:0:0:1	Router7	IF4	10	100	0		Incomplete	
6	2005:0:0:7:0:0:0:0/64	Direct	0:0:0:0:0:0:0:0	Unresolved	Unreachable	10	100	32768		Incomplete	
7	2005:0:0:8:0:0:0:0/64	EBGP	2005:0:0:6:0:0:0:2	Router4	IF11	0	150	0	4 12	Incomplete	[12:144]
8	2005:0:0:9:0:0:0:0/64	RIPng	0:0:0:0:0:0:0:0	Unresolved	Unreachable	10	100	32768		Incomplete	
9	2005:0:0:B:0:0:0:0/64	IBGP	2005:0:0:18:0:0:0:1	Router7	IF4	0	100	0	100 7	Incomplete	
10	2005:0:0:C:0:0:0:0/64	EBGP	2005:0:0:6:0:0:0:2	Router4	IF11	0	150	0	4 12	Incomplete	[12:144]
11	2005:0:0:D:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:1	Router3	IF10	10	100	0	3	Incomplete	
12	2005:0:0:E:0:0:0:0/64	IBGP	2005:0:0:18:0:0:0:1	Router7	IF4	10	100	0		Incomplete	
13	2005:0:0:F:0:0:0:0/64	EBGP	2005:0:0:6:0:0:0:2	Router4	IF11	0	150	0	4 12	Incomplete	[12:144]
14	2005:0:0:10:0:0:0:0/64	EBGP	2005:0:0:6:0:0:0:2	Router4	IF11	10	100	0	4	Incomplete	
15	2005:0:0:18:0:0:0:0/64	RIPng	0:0:0:0:0:0:0:0	Unresolved	Unreachable	10	100	32768		Incomplete	
16											

Figure 3. 15 BGP Routing Table of Router5 in the Community Experiment.

It can be seen that all the routes whose community list contains 12:144 have their local preference set to 150. For example, the prefix 2005:0::0:8.0/64 whose community list is set to [12:144], has a local preference set to 150 by Router5. A route map is defined in Router5 as the following: if the route has the community number 12:144 in its community list, then it assigns the value 150 to its local preference. As a result, the route through Router4 is preferred.

Figure 3.16 shows the convergence activity in this network which depicts similar behavior to the baseline experiment.

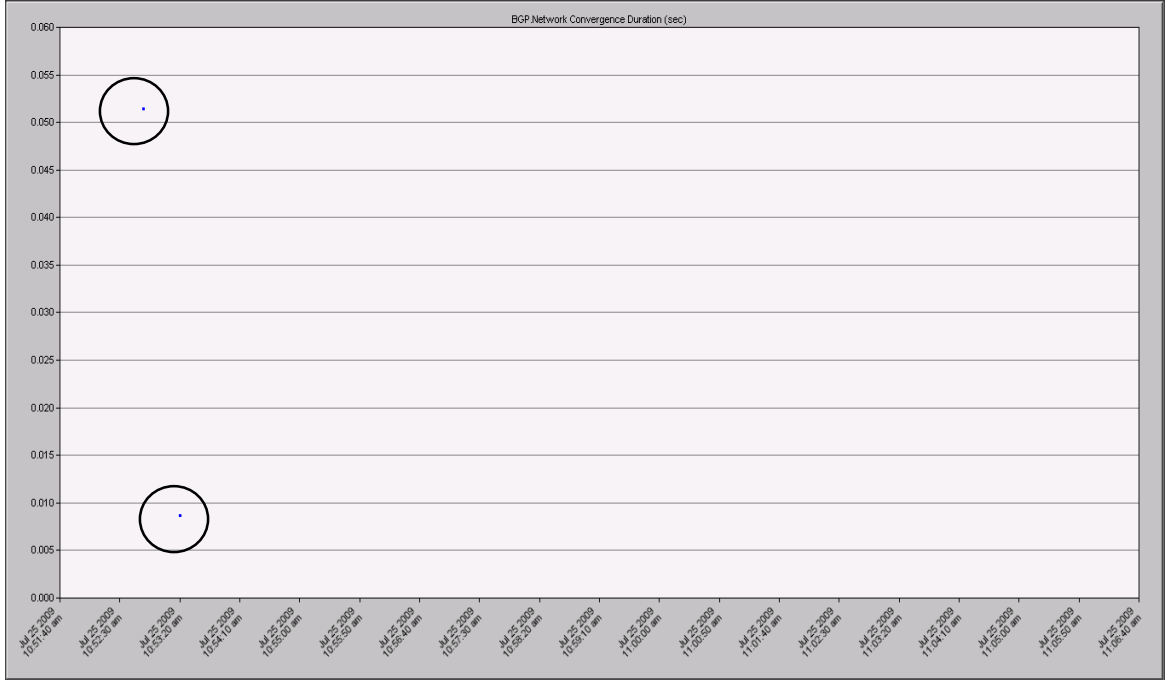


Figure 3. 16 Convergence Activity and Duration of Community Experiment.

3.7.Modification of OPNET Implementation

In the previous approaches we notice that the solutions are applied from the beginning of the simulation. However, we need to change the configuration during the simulation run, but unfortunately OPNET supports limited changes such as failing a node or a link in a specified time. The changes in the configuration that are needed during the simulation run include starting a malicious blocking or applying solutions at specific times. Such changes are not supported by OPNET.

In this section we explain the state model in OPNET and the modifications needed to add the missing features in OPNET that support our experiments. First, we discuss the modifications done to OPNET to add malicious blocking. Then, we explain modifications to the BGP protocol for reconfiguration at a specific time. After that, we give an overview of the BGP process model in OPNET. Finally, we discuss how the

shortening and more specific prefix approaches are to be implemented in OPNET. In all approaches where experiments have been conducted, we use the same network setup used in the baseline scenario that is shown in Figure 3.2.

3.7.1. OPNET Process Model

A process model controls the underlying functionality of the node models. The process models are represented by finite state machines (FSMs) and are created with icons that represent states and lines that represent transitions between states. Operations performed in each state or for a transition are described in embedded C or C++ code blocks.

The states can be forced or unforced (Blocking). The forced states are represented by red color and unforced by the green color. The unforced state is also called idle state. That means it returns control to the simulation kernel after executing its executives. When the simulation starts, the FSM will execute the idle state and will then be ready to transition with the first arriving packet. On the other hand, the forced state does not return control to the simulation kernel, but instead immediately executes the exit executives and transitions to another state.

3.7.2. Building A malicious Router

In this section, we discuss in brief the IP protocol in OPNET. After that, a brief description of the required modifications to the IP protocol to support the evaluated solutions is given. Then, we go through the required modification of OPNET to build the malicious router.

3.7.2.1. Summary of IP Implementation

Each node in OPENT which uses IP has an IP routing module, which contains a dispatcher process that spawns the various routing processes. The ip_dispatch process implements IP routing functions, and fragmentation and reassembly. The ip_dispatch process requires a fixed amount of time to route each packet. Packets are forwarded on a first-come, first-served basis. Figure 3.17 shows the ip_dispatch process model. In our modification we are working with ip_rte_central_cpu process which is a child process of the ip_dispatch process. The ip_rte_central_cpu is responsible for routing all packets from all interfaces in the router. In our modification we are only considering IPv6 packet format.

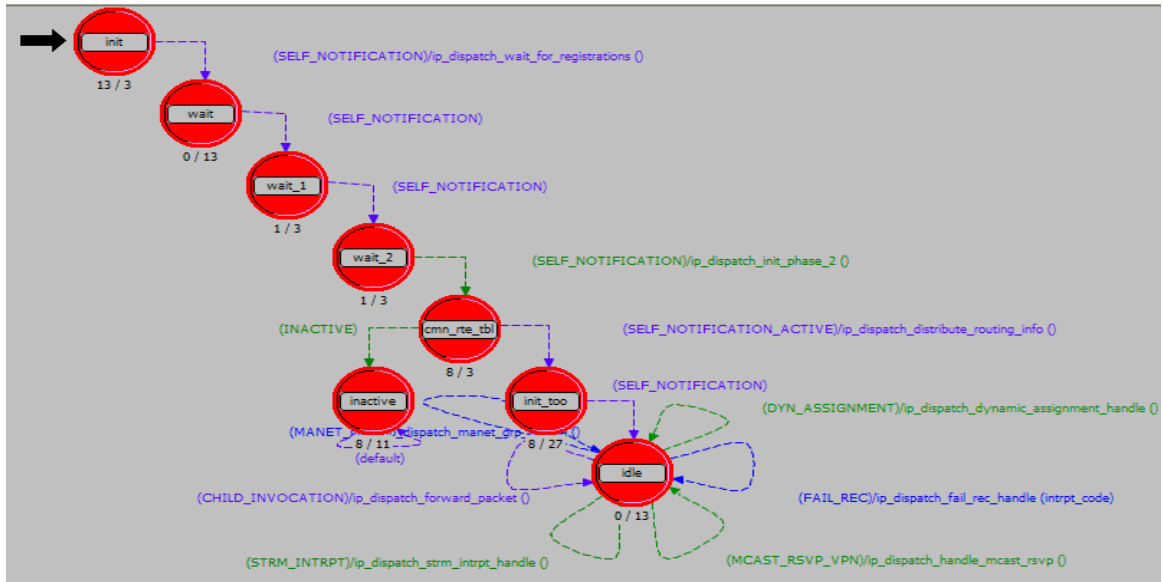


Figure 3. 17 IP_Dispatch Process Model [5].

Figure 3.18 shows the states of the `ip_rte_central_cpu` process model. The `ip_rte_central_cpu_packet_arrival()` is called when a packet arrives. In the case the processing rate was not infinite, the three forced states added to the right of the blocking state ‘`ip_central_cpu`’ add a delay.

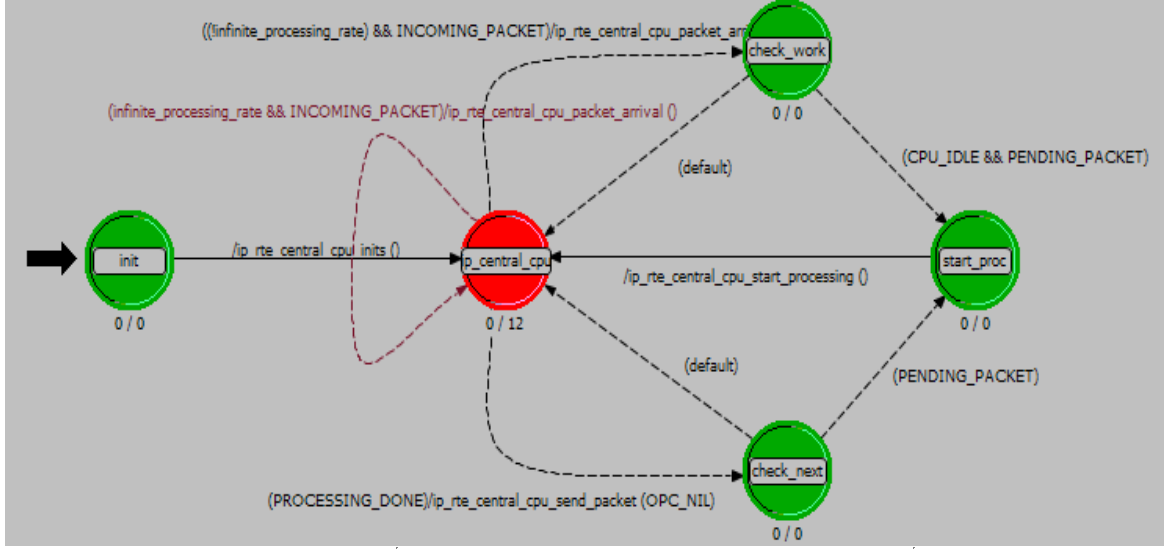


Figure 3.18 `ip_rte_central_cpu` Process Model [5].

3.7.2.2. Malicious Router Implementation

A malicious router acts as a normal router with the exception that it drops the traffic destined to or originating from a specific prefix. In addition, the malicious router continues to advertise to other ASes that it has a route to the blocked prefixes. In order to implement a malicious router a modification to the IP routing module is needed. Moreover, we need to provide an interface for the malicious router to set the blocking prefixes and the time to start the malicious activity.

3.7.2.3. Exact Modifications of the IP Model

The required interface that enables the user to configure the malicious router is added to the `bgp_dispatch` process and is shown in Figure 3.19. Through this interface

the user is able to set the time when the blocking should start and the prefix to be blocked. The blocking process starts by investigating each incoming packet using the `ip_rte_central_cpu_packet_arrival()` method which in turn calls upon the `ip_rte_blackhole_traffic()` that was added by Alrefai [1]. Subsequently, we modified the `ip_rte_blackhole_traffic()` to account for IPv6 traffic. The newly modified method examines each packet against the blocked prefix. Thus, the `ip_rte_central_cpu_packet_arrival()` method calls upon the modified `ip_rte_blackhole_traffic()` method for each incoming packet and decides if this packet belongs to the blocked prefix or not. Refer to [1] to see the activity diagram of the blackholing method. Figure 3.19 shows the malicious router configuration used in the simulation. Accordingly, the malicious activity will start at time 300 and 2008:0:0:8::0 is the blackholed prefix.

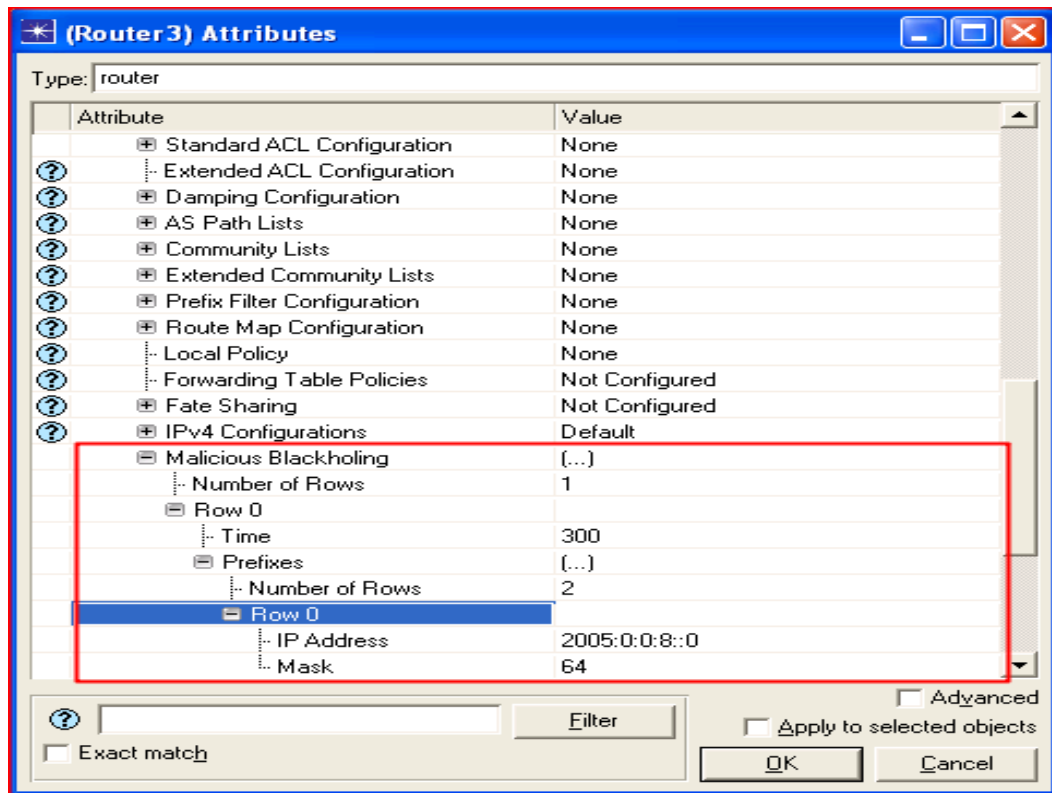


Figure 3. 19 Interface to Configure Malicious Router.

Figure 3.20 shows the throughput between Router2 and Router3, the throughput between Router2 and Router 4, and the dropped traffic of Router3.

The two topmost plots of Figure 3.20 indicate that traffic is being exchanged between Router2 and Router3 for the first 300 seconds. Subsequently, the traffic exchange ceases between Router2 and Router3 because the malicious activity is configured to start at time 300. Moreover, we notice from the last plot in Figure 3.20 the increased packet dropped in Router3. Note that the traffic exchanged between Router2 and Router3 is HTTP traffic which runs on top of TCP. Subsequently, the difference between the number of packets sent and the number of packets dropped, as seen in Figure 3.20 is due to the congestion control feature of TCP.

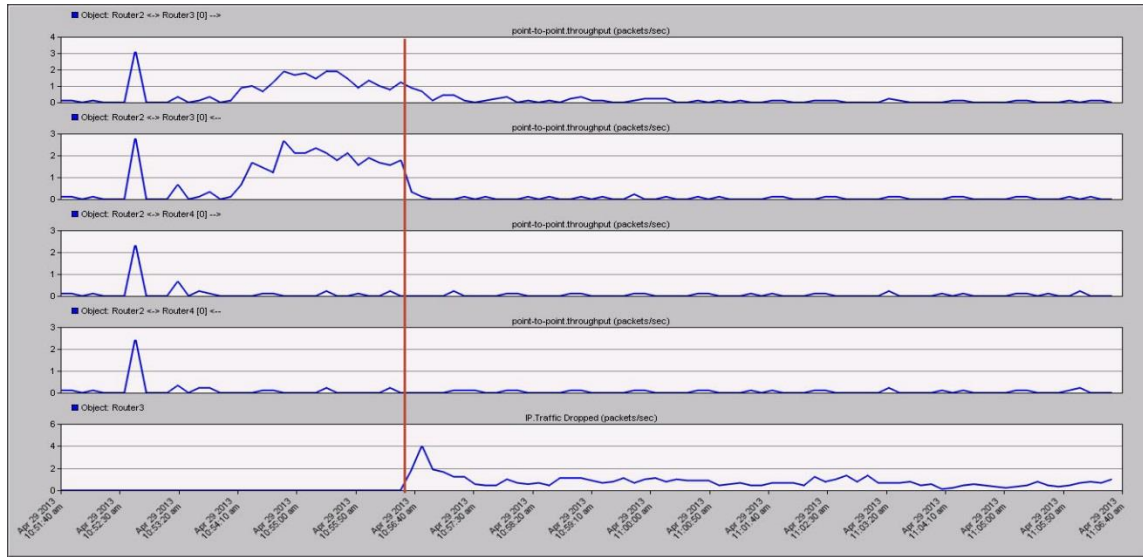


Figure 3.20 Throughput in A malicious Configuration Experiment.

To see that the malicious router is still advertising a path to the local region, Figure 3.21 shows the BGP table of Router5. Figure 3.21 shows that Router3 is the “Next Hop Node” of prefix 2005:0:0:8::0/64 and through the AS-Path [3 12] for Router5.

	Destination	Source Protocol	Next Hop Address	Next Hop Node	Outgoing Interface	MED	Local Preference	Weight	AS Path	Origin
1	2005:0:0:1:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:1	Router3	IF10	0	100	0	3 12	Incomplete
2	2005:0:0:2:0:0:0:0/64	IBGP	2005:0:0:18:0:0:0:1	Router 7	IF4	10	100	0		Incomplete
3	2005:0:0:3:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:1	Router3	IF10	0	100	0	3 12	Incomplete
4	2005:0:0:5:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:1	Router3	IF10	0	100	0	3 12	Incomplete
5	2005:0:0:6:0:0:0:0/64	IBGP	2005:0:0:18:0:0:0:1	Router 7	IF4	10	100	0		Incomplete
6	2005:0:0:7:0:0:0:0/64	Direct	0:0:0:0:0:0:0:0	Unresolved	Unreachable	10	100	32768		Incomplete
7	2005:0:0:8:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:1	Router3	IF10	0	100	0	3 12	Incomplete
8	2005:0:0:9:0:0:0:0/64	RIPng	0:0:0:0:0:0:0:0	Unresolved	Unreachable	10	100	32768		Incomplete
9	2005:0:0:8:0:0:0:0/64	IBGP	2005:0:0:18:0:0:0:1	Router 7	IF4	0	100	0	100 7	Incomplete
10	2005:0:0:C:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:1	Router3	IF10	0	100	0	3 12	Incomplete
11	2005:0:0:D:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:1	Router3	IF10	10	100	0	3	Incomplete
12	2005:0:0:E:0:0:0:0/64	IBGP	2005:0:0:18:0:0:0:1	Router 7	IF4	10	100	0		Incomplete
13	2005:0:0:F:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:1	Router3	IF10	0	100	0	3 12	Incomplete
14	2005:0:0:10:0:0:0:0/64	EBGP	2005:0:0:6:0:0:0:2	Router4	IF11	10	100	0	4	Incomplete
15	2005:0:0:18:0:0:0:0/64	RIPng	0:0:0:0:0:0:0:0	Unresolved	Unreachable	10	100	32768		Incomplete
16										

Figure 3.21 BGP table of Router 5 in Malicious Experiment.

3.8.Building Countermeasures Against a Malicious Act

In this section we discuss the BGP tuning approaches tested in specific time in the simulation after the malicious router has started blackholing the traffic.

First there is a brief introduction to the OPNET BGP module. Then, we will discuss the required modification and added methods for OPNET to control the outgoing traffic. Finally, we will discuss the required modification and added methods for OPNET to control the incoming traffic.

3.8.1. Summary of BGP in OPNET

OPNET models BGP with two processes, the `bgp` process and `bgp_conn` process. Figure 3.22 shows the state diagram of the `bgp` process. The ‘`bgp`’ process is the root process that controls the BGP peering sessions established with neighbors. The process initiates peering connections to all configured neighbors at the specified start time and dispatches any messages from neighbors to the correct `bgp_conn` process. When a BGP message arrives, it also invokes the corresponding child process. A child process is an instance of the `bgp_conn` process model shown in Figure 3.23.

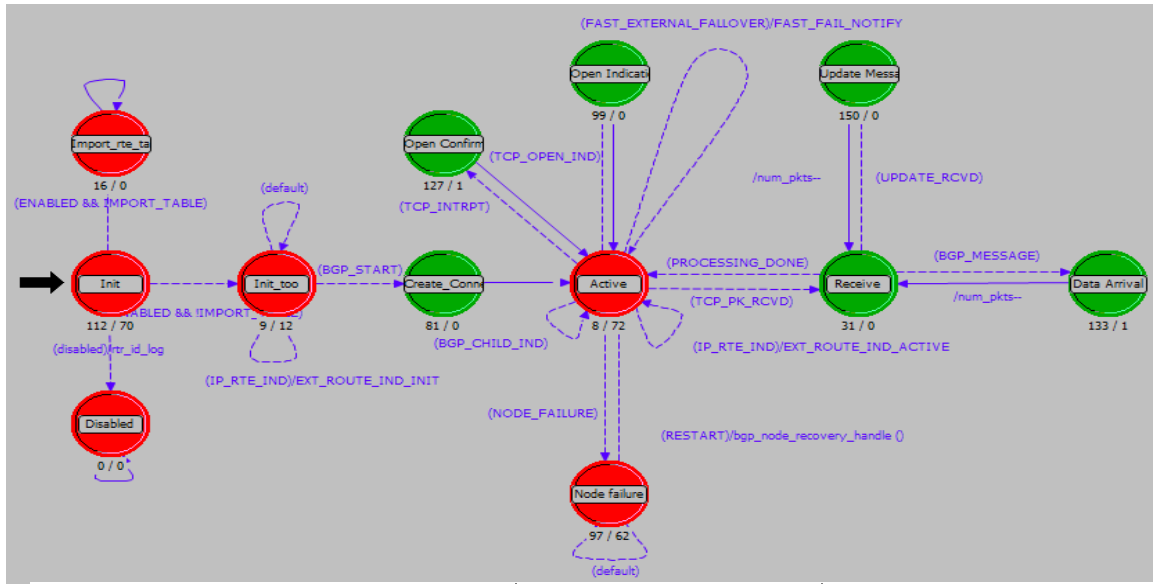


Figure 3. 22 BGP Process [5].

The bgp_conn process model represents the BGP finite state machine. Each BGP peer router communicates with a bgp_conn process which is created for each bgp peer the router communicates with. Initiating, maintaining, and tearing down the BGP and TCP connection processes are the functions that the child process is responsible for.

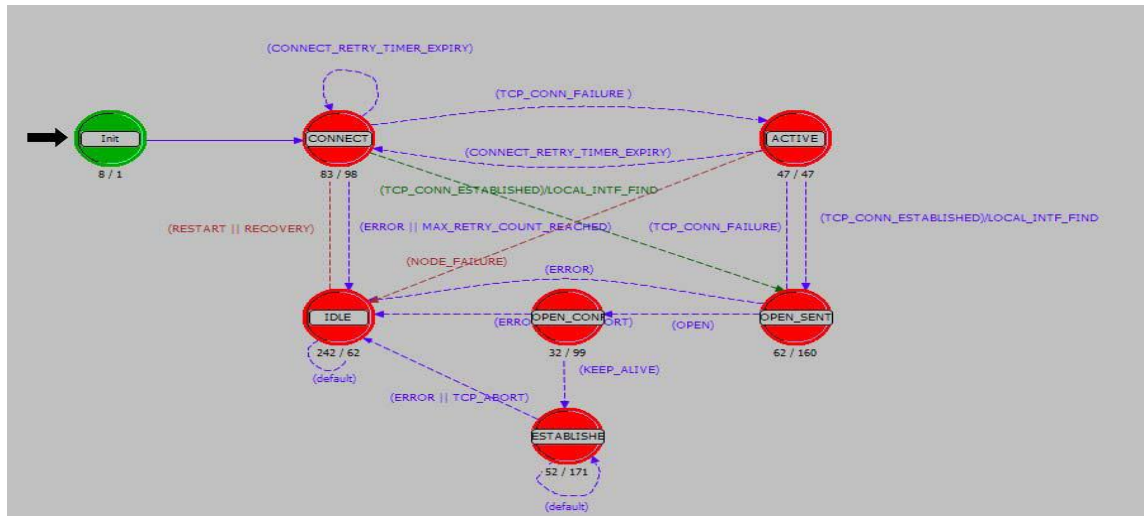


Figure 3. 23 BGP Con Process [5].

3.8.2. BGP Modification for Scheduling Reconfiguration

Each BGP tuning approach has an interface to ease the reconfiguration for the users. Using those interfaces, users could set the time which triggers the reconfiguration for the specific simulated approach. All the reconfiguration requests are saved in a list and at the user specified time those requests will be triggered.

The reconfiguration process is started by reading the time and the reconfiguration request information. Then, a RECONFIGURE event occurs at the time specified in the reconfiguration information. After the RECONFIGURE event is triggered, the direction of the reconfiguration process is determined to reflect whether a reconfiguration of incoming routes or outgoing routes is needed. Finally, the appropriate child process is called to handle the reconfiguration process.

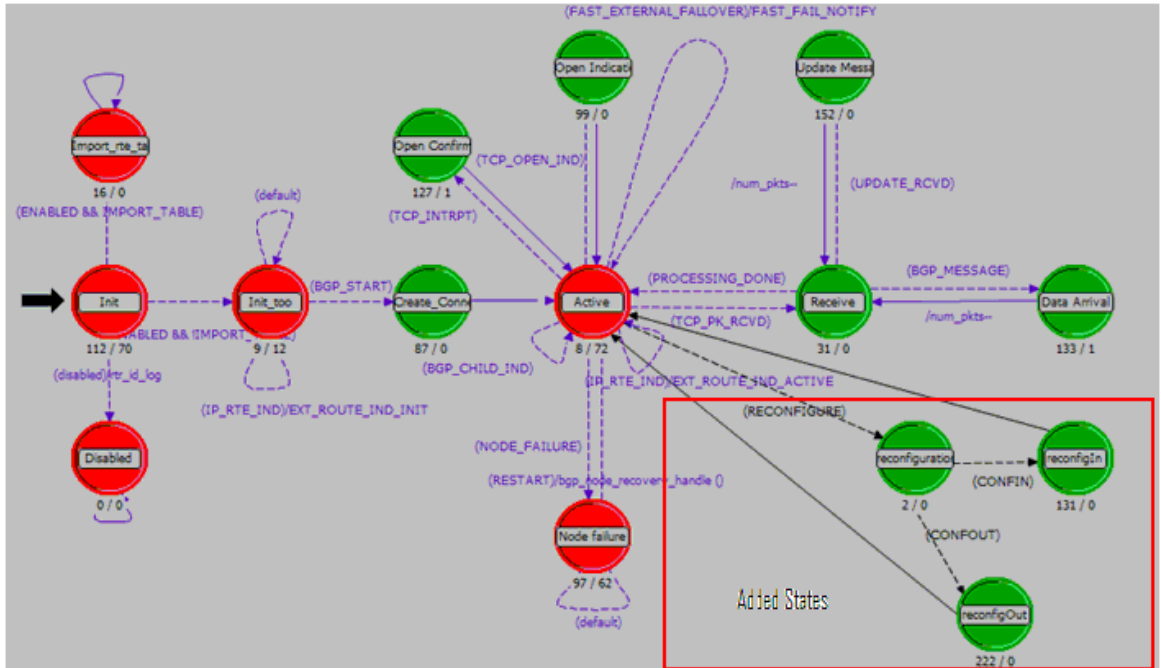


Figure 3. 24 Modified BGP Process Model [1].

Figure 3.24 shows the modified process model of BGP. As described in the reconfiguration process earlier, it shows states in active mode because they will be trigger when the event occurs. The reconfiguration state is used to determine the direction of the update as either 'in' (incoming updates) or 'out' (outgoing updates) so as to triggered the appropriate states reconfigureIn and reconfigureOut.

3.8.2.1. Modification to Control The Outgoing Traffic

Although OPNET supports the concept of Local-Preference, however OPNET does not support changes of the configuration in the middle of the simulation. Subsequently, Alrefai [1] has added this feature by modifying the OPNET code. In our work we modified Alrefai's work to account for IPv6 traffic. Accordingly, Local-Preference is used in order to control the outgoing traffic by assigning a higher Local-Preference value to desired routes.

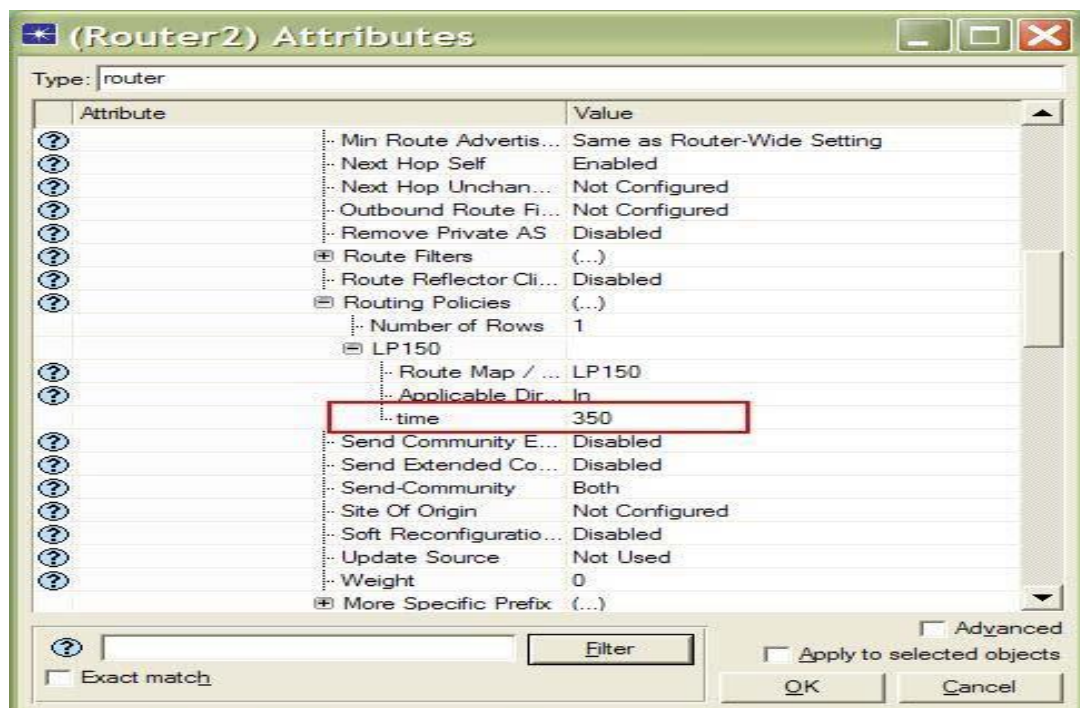


Figure 3. 25 Specification of Time When Applying Route Map.

The simulation scenario is configured to use a malicious router starting at 300, and the local preference route map policy is triggered at 350. Figure 3.25 shows the added interface that enables the user to set the time to trigger the route map.

Figure 3.26 shows the incoming and outgoing traffic between Router2 and Router3, between Router2 and Router4, and dropped traffic at Router3.

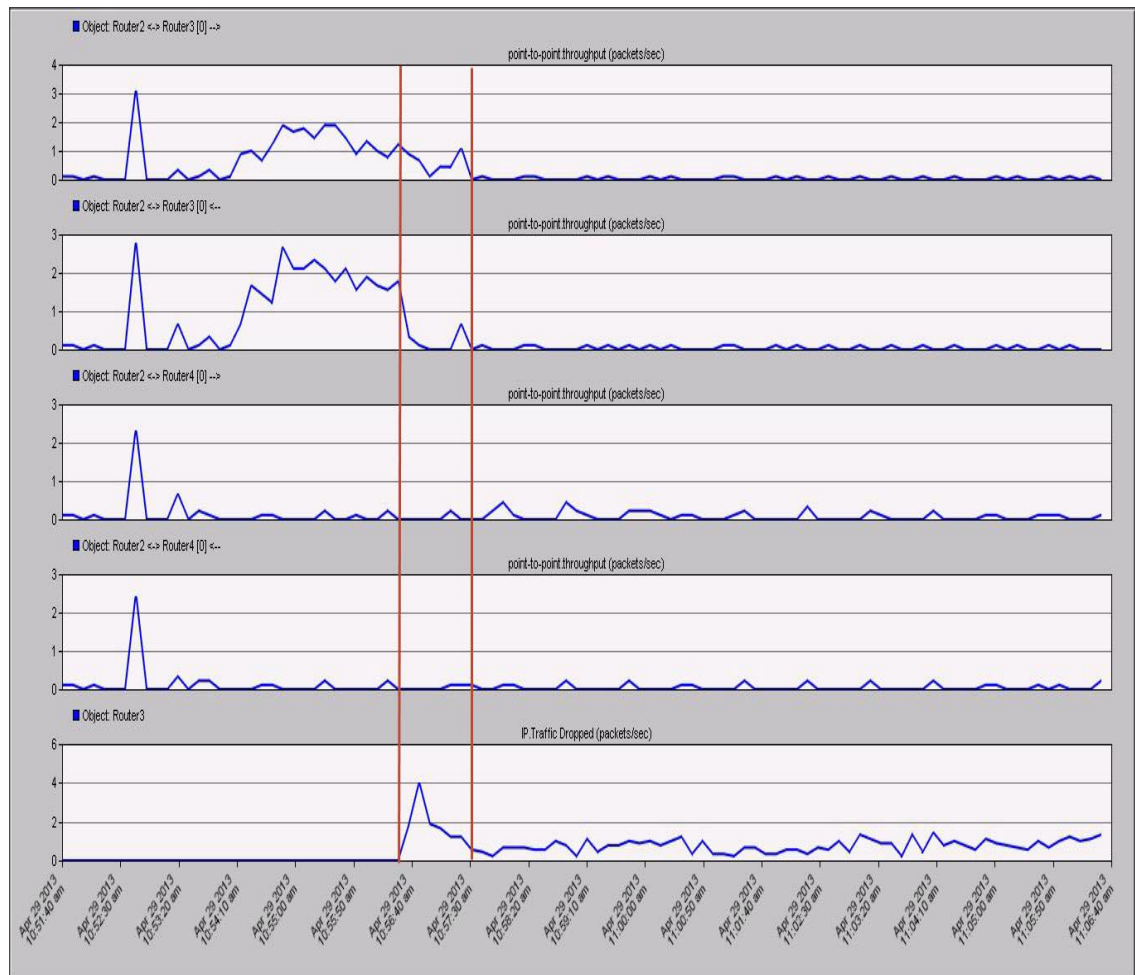


Figure 3. 26 Throughput Traffic After Applying Local-Preference in the Presence of A malicious Router.

As seen in Figure 3.26, it is clear that applying a higher Local-Preference to the route does not show that the outgoing traffic is passing through Router4. To make sure that a change has occurred, we can examine the IP forwarding table of Router2 in Figure 3.27.

	Destination	Source Protocol	Route Preference	Metric	Next Hop Address	Next Hop Node	Outgoing Interface	Outgoing LSP	Insertion Time (secs)
1	2005:0:0:1:0:0:0:0/64	Direct	0	0	2005:0:0:1:0:0:0:1	Router2	IF11	N/A	0.000
2	2005:0:0:1:0:0:0:1/128	Local	0	0	2005:0:0:1:0:0:0:1	Router2	IF11	N/A	0.000
3	2005:0:0:2:0:0:0:0/64	BGP	20	0	2005:0:0:5:0:0:0:2	Router4	IF4	N/A	350.000
4	2005:0:0:3:0:0:0:0/64	Direct	0	0	2005:0:0:3:0:0:0:1	Router2	IF10	N/A	0.000
5	2005:0:0:3:0:0:0:1/128	Local	0	0	2005:0:0:3:0:0:0:1	Router2	IF10	N/A	0.000
6	2005:0:0:5:0:0:0:0/64	Direct	0	0	2005:0:0:5:0:0:0:1	Router2	IF4	N/A	0.000
7	2005:0:0:5:0:0:0:1/128	Local	0	0	2005:0:0:5:0:0:0:1	Router2	IF4	N/A	0.000
8	2005:0:0:6:0:0:0:0/64	BGP	20	0	2005:0:0:5:0:0:0:2	Router4	IF4	N/A	350.000
9	2005:0:0:7:0:0:0:0/64	BGP	20	0	2005:0:0:5:0:0:0:2	Router4	IF4	N/A	350.000
10	2005:0:0:8:0:0:0:0/64	RIPng	120	11	2005:0:0:3:0:0:0:2	Router1	IF10	N/A	5.004
11	2005:0:0:9:0:0:0:0/64	BGP	20	0	2005:0:0:5:0:0:0:2	Router4	IF4	N/A	350.000
12	2005:0:0:B:0:0:0:0/64	BGP	20	0	2005:0:0:5:0:0:0:2	Router4	IF4	N/A	350.000
13	2005:0:0:C:0:0:0:0/64	Direct	0	0	2005:0:0:C:0:0:0:1	Router2	LB0	N/A	0.000
14	2005:0:0:C:0:0:0:1/128	Local	0	0	2005:0:0:C:0:0:0:1	Router2	LB0	N/A	0.000
15	2005:0:0:D:0:0:0:0/64	BGP	20	0	2005:0:0:5:0:0:0:2	Router4	IF4	N/A	350.000
16	2005:0:0:E:0:0:0:0/64	BGP	20	0	2005:0:0:5:0:0:0:2	Router4	IF4	N/A	350.000
17	2005:0:0:F:0:0:0:0/64	RIPng	120	11	2005:0:0:3:0:0:0:2	Router1	IF10	N/A	5.004
18	2005:0:0:10:0:0:0:0/64	BGP	20	10	2005:0:0:5:0:0:0:2	Router4	IF4	N/A	350.000
19	2005:0:0:18:0:0:0:0/64	BGP	20	0	2005:0:0:5:0:0:0:2	Router4	IF4	N/A	350.000
20									

Figure 3. 27 Forwarding Table of Router2 in Local-Preference and Malicious Experiment.

It is clear that at time 350, the prefix to 2005:0:0:B::0/64 is inserted in the table and it has Router4 as the “Next Hop Node”. This route entry was initially through Router3. To see this, Figure 3.27 shows the IP forwarding table for Router2 at time 200.

Figure 3.28 shows that the ‘Next Hop Node’ for the prefix 2005:0:0:B::0/64 before Router3 becomes malicious is Router3 with the local preference value set to 20.

	Destination	Source Protocol	Route Preference	Metric	Next Hop Address	Next Hop Node	Outgoing Interface	Outgoing LSP	Insertion Time (secs)
1	2005:0:0:1:0:0:0:0/64	Direct	0	0	2005:0:0:1:0:0:0:1	Router2	IF11	N/A	0.000
2	2005:0:0:1:0:0:0:1/128	Local	0	0	2005:0:0:1:0:0:0:1	Router2	IF11	N/A	0.000
3	2005:0:0:2:0:0:0:0/64	BGP	20	0	2005:0:0:1:0:0:0:2	Router3	IF11	N/A	100.036
4	2005:0:0:3:0:0:0:0/64	Direct	0	0	2005:0:0:3:0:0:0:1	Router2	IF10	N/A	0.000
5	2005:0:0:3:0:0:0:1/128	Local	0	0	2005:0:0:3:0:0:0:1	Router2	IF10	N/A	0.000
6	2005:0:0:5:0:0:0:0/64	Direct	0	0	2005:0:0:5:0:0:0:1	Router2	IF4	N/A	0.000
7	2005:0:0:5:0:0:0:1/128	Local	0	0	2005:0:0:5:0:0:0:1	Router2	IF4	N/A	0.000
8	2005:0:0:6:0:0:0:0/64	BGP	20	0	2005:0:0:1:0:0:0:2	Router3	IF11	N/A	100.036
9	2005:0:0:7:0:0:0:0/64	BGP	20	0	2005:0:0:1:0:0:0:2	Router3	IF11	N/A	100.037
10	2005:0:0:8:0:0:0:0/64	RIPng	120	11	2005:0:0:3:0:0:0:2	Router1	IF10	N/A	5.004
11	2005:0:0:9:0:0:0:0/64	BGP	20	0	2005:0:0:1:0:0:0:2	Router3	IF11	N/A	100.037
12	2005:0:0:8:0:0:0:0/64	BGP	20	0	2005:0:0:1:0:0:0:2	Router3	IF11	N/A	130.037
13	2005:0:0:C:0:0:0:0/64	Direct	0	0	2005:0:0:C:0:0:0:1	Router2	LB0	N/A	0.000
14	2005:0:0:C:0:0:0:1/128	Local	0	0	2005:0:0:C:0:0:0:1	Router2	LB0	N/A	0.000
15	2005:0:0:D:0:0:0:0/64	BGP	20	10	2005:0:0:1:0:0:0:2	Router3	IF11	N/A	70.031
16	2005:0:0:E:0:0:0:0/64	BGP	20	0	2005:0:0:1:0:0:0:2	Router3	IF11	N/A	100.036
17	2005:0:0:F:0:0:0:0/64	RIPng	120	11	2005:0:0:3:0:0:0:2	Router1	IF10	N/A	5.004
18	2005:0:0:10:0:0:0:0/64	BGP	20	10	2005:0:0:5:0:0:0:2	Router4	IF4	N/A	70.028
19	2005:0:0:18:0:0:0:0/64	BGP	20	0	2005:0:0:1:0:0:0:2	Router3	IF11	N/A	100.036
20									

Figure 3. 28 IP Forwarding Table of Router2 at Time 200 in the Local-Preference and Malicious Experiment.

Figure 3.29 shows the BGP routing table at time 2000. As shown, Router4 is the next hop to LAN_East. However, we observe from Figure 3.26 that there is no outgoing traffic after Router3 becomes malicious and this is because of the application type used. LAN_East works as an HTTP client that requests services from the HTTP servers hosted at the LAN_West. The HTTP request is sent from LAN_East to LAN_West and LAN_West will send a reply through Router3. However, after Router3 becomes malicious, the subsequent requests will not reach LAN_West because of the malicious activity and the Local-Preference has no effect on the incoming traffic.

	Destination	Source Protocol	Next Hop Address	Next Hop Node	Outgoing Interface	MED	Local Preference	Weight	AS Path	Origin
1	2005:0:0:1:0:0:0:0/64	IBGP	2005:0:0:F:0:0:0:1	Router1	IF10	10	100	0		Incomplete
2	2005:0:0:2:0:0:0:0/64	EBGP	2005:0:0:5:0:0:0:2	Router4	IF4	0	150	0	4 56	Incomplete
3	2005:0:0:3:0:0:0:0/64	Direct	0:0:0:0:0:0:0:0	Unresolved	Unreachable	10	100	32768		Incomplete
4	2005:0:0:5:0:0:0:0/64	IBGP	2005:0:0:F:0:0:0:1	Router1	IF10	10	100	0		Incomplete
5	2005:0:0:6:0:0:0:0/64	EBGP	2005:0:0:5:0:0:0:2	Router4	IF4	0	150	0	4 56	Incomplete
6	2005:0:0:7:0:0:0:0/64	EBGP	2005:0:0:5:0:0:0:2	Router4	IF4	0	150	0	4 56	Incomplete
7	2005:0:0:8:0:0:0:0/64	RIPng	0:0:0:0:0:0:0:0	Unresolved	Unreachable	10	100	32768		Incomplete
8	2005:0:0:9:0:0:0:0/64	EBGP	2005:0:0:5:0:0:0:2	Router4	IF4	0	150	0	4 56	Incomplete
9	2005:0:0:8:0:0:0:0/64	EBGP	2005:0:0:5:0:0:0:2	Router4	IF4	0	150	0	4 56 100 7	Incomplete
10	2005:0:0:C:0:0:0:0/64	IBGP	2005:0:0:F:0:0:0:1	Router1	IF10	10	100	0		Incomplete
11	2005:0:0:D:0:0:0:0/64	EBGP	2005:0:0:5:0:0:0:2	Router4	IF4	0	150	0	4 3	Incomplete
12	2005:0:0:E:0:0:0:0/64	EBGP	2005:0:0:5:0:0:0:2	Router4	IF4	0	150	0	4 56	Incomplete
13	2005:0:0:F:0:0:0:0/64	RIPng	0:0:0:0:0:0:0:0	Unresolved	Unreachable	10	100	32768		Incomplete
14	2005:0:0:10:0:0:0:0/64	EBGP	2005:0:0:5:0:0:0:2	Router4	IF4	10	150	0	4	Incomplete
15	2005:0:0:18:0:0:0:0/64	EBGP	2005:0:0:5:0:0:0:2	Router4	IF4	0	150	0	4 56	Incomplete
16										

Figure 3. 29 Routing Table of Router2 in Local-Preference and Malicious Experiment

3.8.2.2. Control of Incoming Traffic: The Use of Community

The use of the community is one of the possible solutions that can be applied to control the incoming traffic that is supported by OPNET when dealing with a malicious ISP. On the other hand, OPNET does not support the configuration of community at a specific time. So, Alrefai [1] modified OPNET to use the community while the simulation is running. Subsequently, we modified Alrefai's work to consider IPv6 traffic.

When using the community scenario to control the incoming traffic, we configure Router2 to send community number 12:144 to Router4 at time 350 time. Subsequently, Router4 will advertise this community number to Router5 which will assign higher local preference to routes with that community number, and forces the traffic to go through the non-malicious ISP. Figure 3.30 shows the throughput between

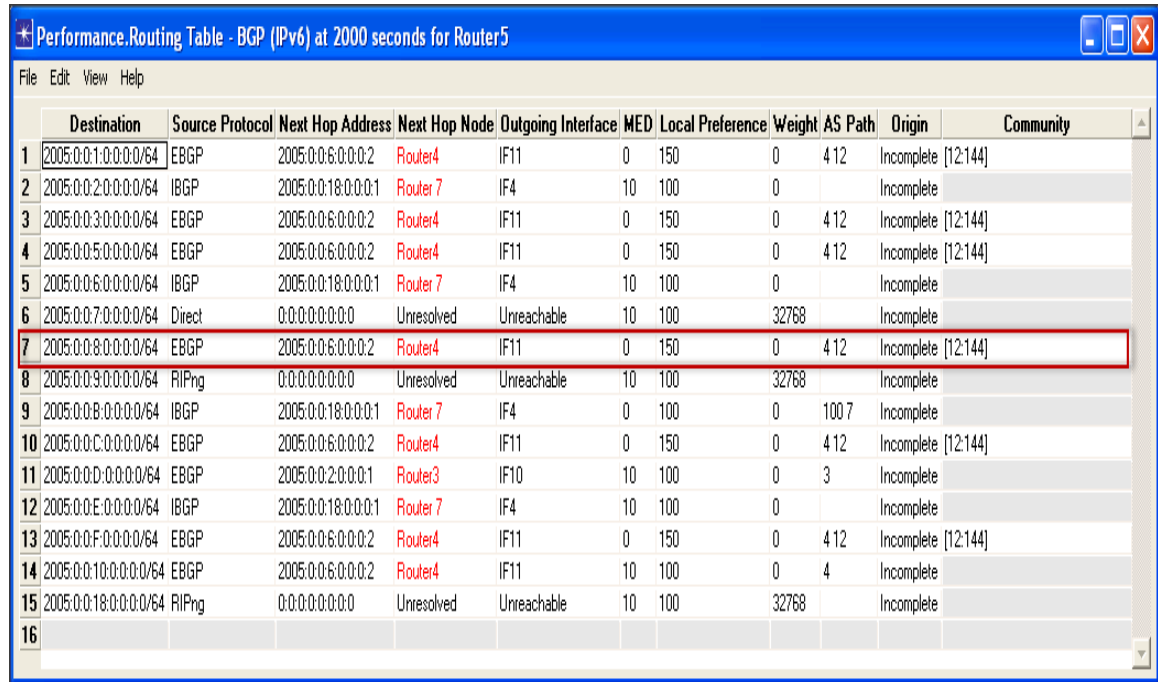
Router2 and Router3, and between Router2 and Router4 in both directions. In addition, the figure shows the packets dropped due to the malicious act of Router3 that started at time 300.

Figure 3.30 shows that the traffic between Router2 and Router3 ceases after Router3 became malicious. After applying the solution, the traffic is exchanged between Router2 and Router4. The bottom part of Figure 3.30 shows the dropped traffic in Router3 which happens after Router3 became malicious and before applying the solution.



Figure 3. 30 Throughput between Router2 and Router3, Router4 and Packet drop of Router3.

Figure 3.31 shows the BGP routing table of Router5 at the end of the simulation. The Router5 routing table in Figure 3.31 shows the advertised community [12:144] that was sent by Router2 is associated with 2005:0:0:8::0/64. Accordingly, Router4 is set as the ‘Next Hop Node’ for prefix 2005:0:0:8::0/64 with a Local-Preference set to 150.



	Destination	Source Protocol	Next Hop Address	Next Hop Node	Outgoing Interface	MED	Local Preference	Weight	AS Path	Origin	Community
1	2005:0:0:1:0:0:0:0/64	EBGP	2005:0:0:6:0:0:0:2	Router4	IF11	0	150	0	412	Incomplete	[12:144]
2	2005:0:0:2:0:0:0:0/64	IBGP	2005:0:0:18:0:0:0:1	Router7	IF4	10	100	0		Incomplete	
3	2005:0:0:3:0:0:0:0/64	EBGP	2005:0:0:6:0:0:0:2	Router4	IF11	0	150	0	412	Incomplete	[12:144]
4	2005:0:0:5:0:0:0:0/64	EBGP	2005:0:0:6:0:0:0:2	Router4	IF11	0	150	0	412	Incomplete	[12:144]
5	2005:0:0:6:0:0:0:0/64	IBGP	2005:0:0:18:0:0:0:1	Router7	IF4	10	100	0		Incomplete	
6	2005:0:0:7:0:0:0:0/64	Direct	0:0:0:0:0:0:0:0	Unresolved	Unreachable	10	100	32768		Incomplete	
7	2005:0:0:8:0:0:0:0/64	EBGP	2005:0:0:6:0:0:0:2	Router4	IF11	0	150	0	412	Incomplete	[12:144]
8	2005:0:0:9:0:0:0:0/64	RIPng	0:0:0:0:0:0:0:0	Unresolved	Unreachable	10	100	32768		Incomplete	
9	2005:0:0:8:0:0:0:0/64	IBGP	2005:0:0:18:0:0:0:1	Router7	IF4	0	100	0	100 7	Incomplete	
10	2005:0:0:C:0:0:0:0/64	EBGP	2005:0:0:6:0:0:0:2	Router4	IF11	0	150	0	412	Incomplete	[12:144]
11	2005:0:0:D:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:1	Router3	IF10	10	100	0	3	Incomplete	
12	2005:0:0:E:0:0:0:0/64	IBGP	2005:0:0:18:0:0:0:1	Router7	IF4	10	100	0		Incomplete	
13	2005:0:0:F:0:0:0:0/64	EBGP	2005:0:0:6:0:0:0:2	Router4	IF11	0	150	0	412	Incomplete	[12:144]
14	2005:0:0:10:0:0:0:0/64	EBGP	2005:0:0:6:0:0:0:2	Router4	IF11	10	100	0	4	Incomplete	
15	2005:0:0:18:0:0:0:0/64	RIPng	0:0:0:0:0:0:0:0	Unresolved	Unreachable	10	100	32768		Incomplete	
16											

Figure 3. 31 BGP Routing table of Router5 in Malicious and Community Experiment.

On the other hand, as shown in Figure 3.32, Router5 routing table at time 71seconds does not show any community number and the ‘Next Hop Node’ to the prefix 2005:0:0:8::0/64 is Router3 since the solution has not been applied yet.

Performance.Routing Table - BGP (IPv6) at 71 seconds for Router5											
	Destination	Source Protocol	Next Hop Address	Next Hop Node	Outgoing Interface	MED	Local Preference	Weight	AS Path	Origin	Community
1	2005:0:0:1:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:1	Router3	IF10	0	100	0	312	Incomplete	
2	2005:0:0:2:0:0:0:0/64	IBGP	2005:0:0:18:0:0:0:1	Router 7	IF4	10	100	0		Incomplete	
3	2005:0:0:3:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:1	Router3	IF10	0	100	0	312	Incomplete	
4	2005:0:0:5:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:1	Router3	IF10	0	100	0	312	Incomplete	
5	2005:0:0:6:0:0:0:0/64	IBGP	2005:0:0:18:0:0:0:1	Router 7	IF4	10	100	0		Incomplete	
6	2005:0:0:7:0:0:0:0/64	Direct	0:0:0:0:0:0:0:0	Unresolved	Unreachable	10	100	32768		Incomplete	
7	2005:0:0:8:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:1	Router3	IF10	0	100	0	312	Incomplete	
8	2005:0:0:9:0:0:0:0/64	RIPng	0:0:0:0:0:0:0:0	Unresolved	Unreachable	10	100	32768		Incomplete	
9	2005:0:0:C:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:1	Router3	IF10	0	100	0	312	Incomplete	
10	2005:0:0:D:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:1	Router3	IF10	10	100	0	3	Incomplete	
11	2005:0:0:E:0:0:0:0/64	IBGP	2005:0:0:18:0:0:0:1	Router 7	IF4	10	100	0		Incomplete	
12	2005:0:0:F:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:1	Router3	IF10	0	100	0	312	Incomplete	
13	2005:0:0:10:0:0:0:0/64	EBGP	2005:0:0:6:0:0:0:2	Router4	IF11	10	100	0	4	Incomplete	
14	2005:0:0:18:0:0:0:0/64	RIPng	0:0:0:0:0:0:0:0	Unresolved	Unreachable	10	100	32768		Incomplete	
15											

Figure 3.32 BGP Routing table of Router5 in Malicious and Community Experiment at Time 71 Seconds.

Figure 3.33 shows the convergence activity and duration of the community and approval. The network takes about 0.026 seconds to converge after the use of community is applied and is represented as a third point in Figure 3.33.

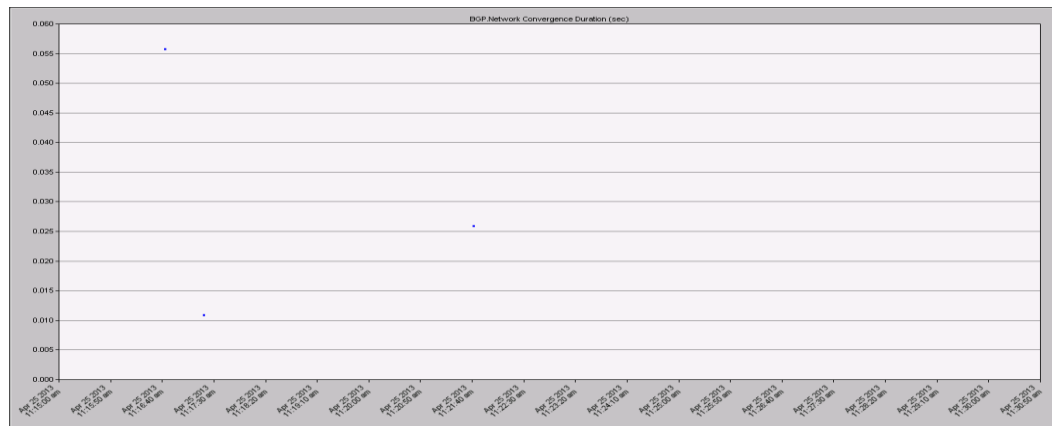


Figure 3.33 Convergence Activity in Community and Malicious.

3.8.2.3. AS-Path Shortening

Shortening is one of the approaches that can be used to bypass the Internet access denial problem that OPNET did not support. Implementing shortening in OPNET was introduced by Alrefai [1].

The shortening approach requires an agreement between two routers (Regional and non-malicious ISP). Shortening the route means that the ISP advertises the route to the regional AS with an AS-Path that contains only the AS number of the ISP while eliminating the AS number of the regional AS. Hence, the shortening approach works by sending update messages from the regional router to the non-malicious ISP, then the ISP will shorten that route. As a result, the advertised route from the non-malicious ISP will be shorter by one and it will be more preferred than the route received from the malicious ISP. In the shortening scenario, we use shortening and local preference together at time 350 to control the incoming and outgoing traffic whereas the malicious activity starts at time 300.

Figure 3.34 shows the throughput between Router2 and Router3, and between Router2 and Router4 in both directions in addition to the traffic dropped at Router3.



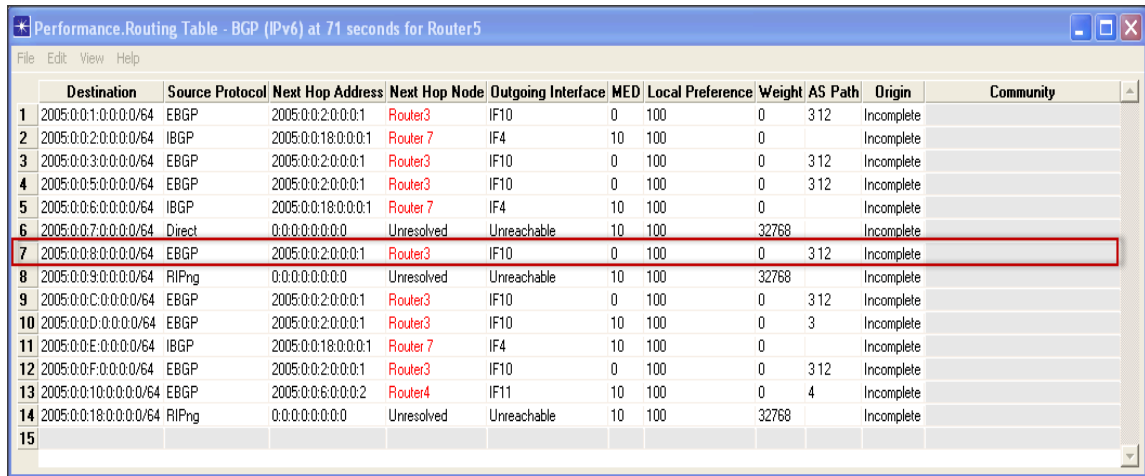
Figure 3. 34 Throughput Between Router2 and Routers3 and Router4 and Dropped Traffic of Router3.

Figure 3.34 shows the incoming and outgoing traffic passing through Router4 after applying the solution while there is a small amount of the dropped packets during the period from the start of the malicious activity until the start of the solution. Moreover, Figure 3.35 shows the Router5 BGP routing table at the end of the simulation.

	Destination	Source Protocol	Next Hop Address	Next Hop Node	Outgoing Interface	MED	Local Preference	Weight	AS Path	Origin	Community
1	2005:0:0:1:0:0:0:0/64	EBGP	2005:0:0:6:0:0:0:2	Router4	IF11	0	100	0	4	Incomplete	[12:145]
2	2005:0:0:2:0:0:0:0/64	IBGP	2005:0:0:18:0:0:0:1	Router7	IF4	10	100	0		Incomplete	
3	2005:0:0:3:0:0:0:0/64	EBGP	2005:0:0:6:0:0:0:2	Router4	IF11	0	100	0	4	Incomplete	[12:145]
4	2005:0:0:5:0:0:0:0/64	EBGP	2005:0:0:6:0:0:0:2	Router4	IF11	0	100	0	4	Incomplete	[12:145]
5	2005:0:0:6:0:0:0:0/64	IBGP	2005:0:0:18:0:0:0:1	Router7	IF4	10	100	0		Incomplete	
6	2005:0:0:7:0:0:0:0/64	Direct	0:0:0:0:0:0:0:0	Unresolved	Unreachable	10	100	32768		Incomplete	
7	2005:0:0:8:0:0:0:0/64	EBGP	2005:0:0:6:0:0:0:2	Router4	IF11	0	100	0	4	Incomplete	[12:145]
8	2005:0:0:9:0:0:0:0/64	RIPng	0:0:0:0:0:0:0:0	Unresolved	Unreachable	10	100	32768		Incomplete	
9	2005:0:0:8:0:0:0:0/64	IBGP	2005:0:0:18:0:0:0:1	Router7	IF4	0	100	0	100 7	Incomplete	
10	2005:0:0:C:0:0:0:0/64	EBGP	2005:0:0:6:0:0:0:2	Router4	IF11	0	100	0	4	Incomplete	[12:145]
11	2005:0:0:D:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:1	Router3	IF10	10	100	0	3	Incomplete	
12	2005:0:0:E:0:0:0:0/64	IBGP	2005:0:0:18:0:0:0:1	Router7	IF4	10	100	0		Incomplete	
13	2005:0:0:F:0:0:0:0/64	EBGP	2005:0:0:6:0:0:0:2	Router4	IF11	0	100	0	4	Incomplete	[12:145]
14	2005:0:0:10:0:0:0:0/64	EBGP	2005:0:0:6:0:0:0:2	Router4	IF11	10	100	0	4	Incomplete	
15	2005:0:0:18:0:0:0:0/64	RIPng	0:0:0:0:0:0:0:0	Unresolved	Unreachable	10	100	32768		Incomplete	
16											

Figure 3. 35 BGP Routing Table of Router5 in Shortening, Local-Pref, Malicious Experiment.

Figure 3.35 shows that Router5 prefers Router4 for prefix 2005:0:0:8::0/64 because it has the shortest AS-Path length. Furthermore, Figure 3.36 shows the routing table of Router5 at time 71 before applying the malicious activity and the shortening solution. The figure shows that for the same prefix, Router3 was chosen as the next hop which results in an AS-Path length of 2.



	Destination	Source Protocol	Next Hop Address	Next Hop Node	Outgoing Interface	MED	Local Preference	Weight	AS Path	Origin	Community
1	2005:0:0:1:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:1	Router3	IF10	0	100	0	3 12	Incomplete	
2	2005:0:0:2:0:0:0:0/64	IBGP	2005:0:0:18:0:0:0:1	Router 7	IF4	10	100	0		Incomplete	
3	2005:0:0:3:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:1	Router3	IF10	0	100	0	3 12	Incomplete	
4	2005:0:0:5:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:1	Router3	IF10	0	100	0	3 12	Incomplete	
5	2005:0:0:6:0:0:0:0/64	IBGP	2005:0:0:18:0:0:0:1	Router 7	IF4	10	100	0		Incomplete	
6	2005:0:0:7:0:0:0:0/64	Direct	0:0:0:0:0:0:0:0	Unresolved	Unreachable	10	100	32768		Incomplete	
7	2005:0:0:8:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:1	Router3	IF10	0	100	0	3 12	Incomplete	
8	2005:0:0:9:0:0:0:0/64	RIPng	0:0:0:0:0:0:0:0	Unresolved	Unreachable	10	100	32768		Incomplete	
9	2005:0:0:C:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:1	Router3	IF10	0	100	0	3 12	Incomplete	
10	2005:0:0:D:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:1	Router3	IF10	10	100	0	3	Incomplete	
11	2005:0:0:E:0:0:0:0/64	IBGP	2005:0:0:18:0:0:0:1	Router 7	IF4	10	100	0		Incomplete	
12	2005:0:0:F:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:1	Router3	IF10	0	100	0	3 12	Incomplete	
13	2005:0:0:10:0:0:0:0/64	EBGP	2005:0:0:6:0:0:0:2	Router4	IF11	10	100	0	4	Incomplete	
14	2005:0:0:18:0:0:0:0/64	RIPng	0:0:0:0:0:0:0:0	Unresolved	Unreachable	10	100	32768		Incomplete	
15											

Figure 3. 36 BGP Routing Table of Router5 in As- Path Shortening and Local-Preference, Malicious Experiment.

Figure 3.37 shows the convergence activity of the shortening scenario. We notice that a third point is added which shows the change that happened in the middle of the simulation at time 350 when the shortening solution was applied. It takes about 0.021 seconds for the shortening solution to converge.

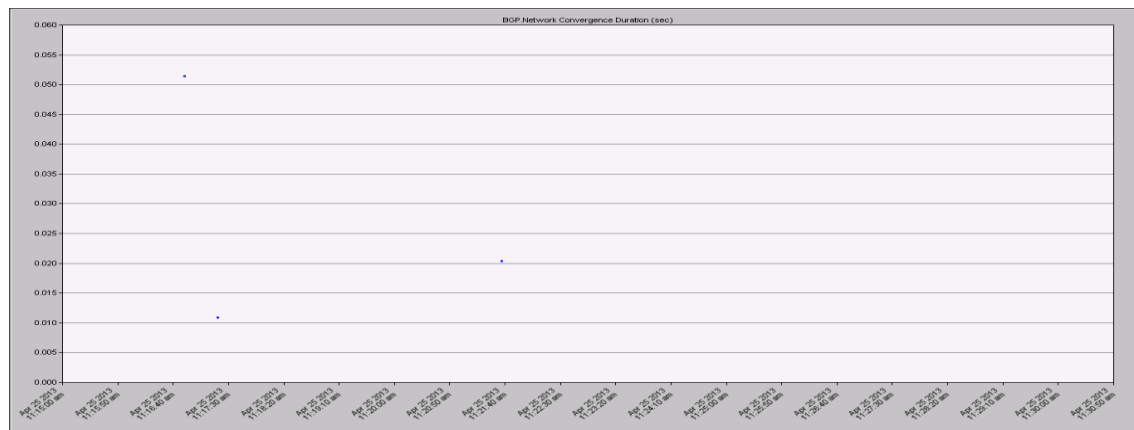


Figure 3. 37 Convergence activity and duration for shortening experiment.

3.8.2.4. More Specific Prefixes

OPNET gives the user the ability to send prefixes which are not in the routing table. On the other hand, it sends these prefixes to all neighbors. A more specific prefix approach works by sending the more specific prefixes to a specific neighbor. Accordingly, we modified the more specific approach that was implemented by Alrefai [1] to account for IPv6 prefixes and traffic.



Figure 3.38 Incoming and Outgoing Traffic of Router2 in More Specific , Local Preference, Malicious Experiment.

Figure 3.38 shows the incoming and outgoing traffic passing through Router4 after applying the solution. Furthermore, the figure shows that there is a small amount of dropped packets during the period between the start of the malicious activity and the start of the solution.

As shown in Figure 3.39, the newly added prefixes are 2005:0:0:8::2/128 and 2005:0:0:8::128/128. Moreover, the default prefix 2005:0:0:8::0/64 still exists in the

Performance.Routing Table - BGP (IPv6) at 2000 seconds for Router5										
	Destination	Source Protocol	Next Hop Address	Next Hop Node	Outgoing Interface	MED	Local Preference	Weight	AS Path	Origin
1	2005:0:0:1:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:1	Router3	IF10	0	100	0	3 12	Incomplete
2	2005:0:0:2:0:0:0:0/64	IBGP	2005:0:0:18:0:0:0:1	Router7	IF4	10	100	0		Incomplete
3	2005:0:0:3:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:1	Router3	IF10	0	100	0	3 12	Incomplete
4	2005:0:0:5:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:1	Router3	IF10	0	100	0	3 12	Incomplete
5	2005:0:0:6:0:0:0:0/64	IBGP	2005:0:0:18:0:0:0:1	Router7	IF4	10	100	0		Incomplete
6	2005:0:0:7:0:0:0:0/64	Direct	0:0:0:0:0:0:0:0	Unresolved	Unreachable	10	100	32768		Incomplete
7	2005:0:0:8:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:1	Router3	IF10	0	100	0	3 12	Incomplete
8	2005:0:0:8:0:0:0:2/128	EBGP	2005:0:0:6:0:0:0:2	Router4	IF11	0	100	0	4 12	Incomplete
9	2005:0:0:8:0:0:0:128/128	EBGP	2005:0:0:6:0:0:0:2	Router4	IF11	0	100	0	4 12	Incomplete
10	2005:0:0:9:0:0:0:0/64	RIPng	0:0:0:0:0:0:0:0	Unresolved	Unreachable	10	100	32768		Incomplete
11	2005:0:0:8:0:0:0:0/64	IBGP	2005:0:0:18:0:0:0:1	Router7	IF4	0	100	0	100 7	Incomplete
12	2005:0:0:C:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:1	Router3	IF10	0	100	0	3 12	Incomplete
13	2005:0:0:D:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:1	Router3	IF10	10	100	0	3	Incomplete
14	2005:0:0:E:0:0:0:0/64	IBGP	2005:0:0:18:0:0:0:1	Router7	IF4	10	100	0		Incomplete
15	2005:0:0:F:0:0:0:0/64	EBGP	2005:0:0:2:0:0:0:1	Router3	IF10	0	100	0	3 12	Incomplete
16	2005:0:0:10:0:0:0:0/64	EBGP	2005:0:0:6:0:0:0:2	Router4	IF11	10	100	0	4	Incomplete
17	2005:0:0:18:0:0:0:0/64	RIPng	0:0:0:0:0:0:0:0	Unresolved	Unreachable	10	100	32768		Incomplete
18										

Figure 3. 39 Routing table for Router5 in More Specific Experiment.

table. Since BGP uses longest prefix matching, those newly added prefixes will be more preferred and the incoming traffic will pass through Router4.

Figure 3.40 shows the convergence activity of the more specific prefixes scenario. The third point in Figure 3.40 represents the convergence of the network for the more specific prefixes solution. It takes about 0.020 seconds for the network to converge.

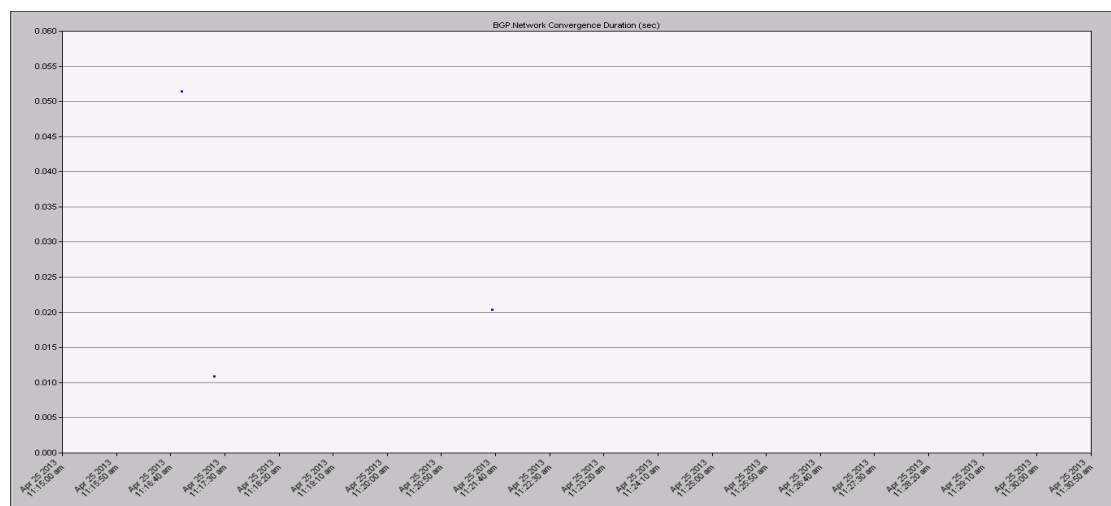


Figure 3. 40 Convergence Activity and Duration of More Specific, Local Preference, and Malicious Experiment.

CHAPTER 4

PERFORMANCE EVALUATION OF BGP TUNING TECHNIQUES TO CIRCUMVENT MALICIOUS ACT

4.1. Introduction

In this chapter, we discuss and illustrate the performance evaluation of the BGP-based solutions that were proposed by Alrefai [1]. We have simulated the BGP-based solutions with different configurations such as background traffic loads, Internet delay and traffic types. Figure 4.1 shows the network topology that is used in the performance evaluation which is the same topology used in the implementation and validation chapter.

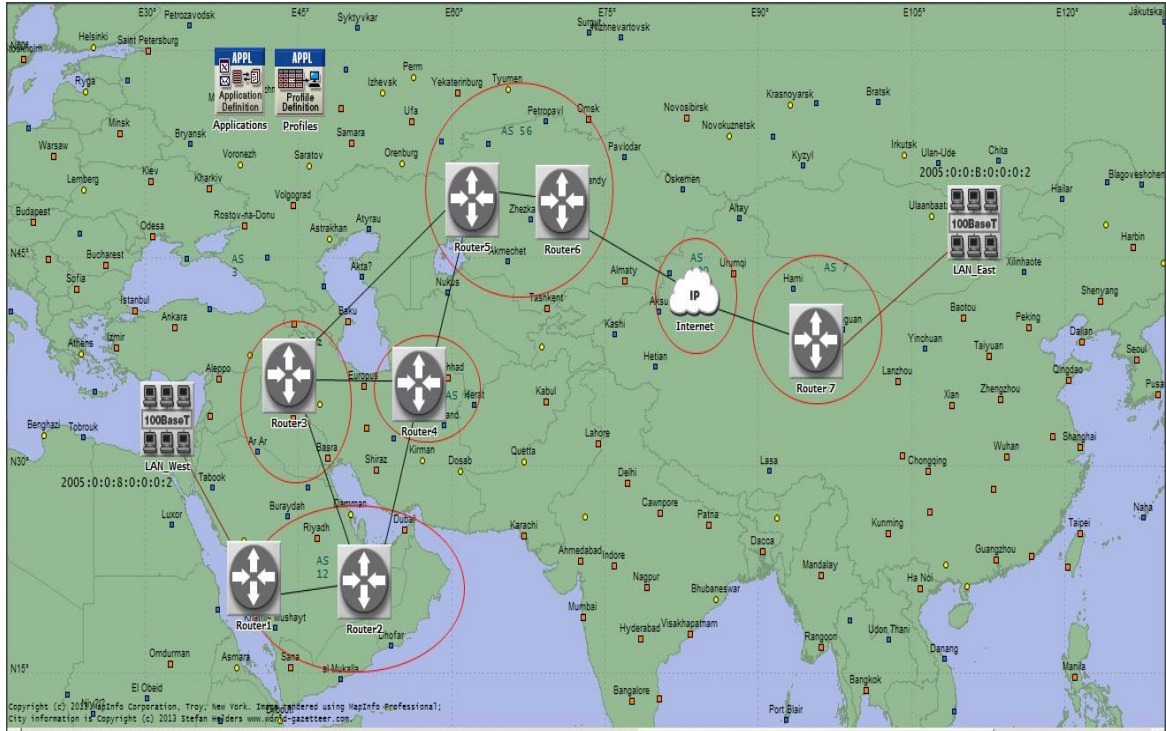


Figure 4.1 Evaluation Network Setup.

Similar to the work done by AlRefai [1], the data rate of the link connecting the routers have been reduced from 44.736 Mbps to 1.544 Mbps. The reason behind the reduction in the data rate is to allow for the study of the effect of the network load within a reasonable simulation time. IP_cloud is used to model the Internet delay which follows an exponential distribution with a mean of either 0.1 seconds or 5 seconds. Both links from the IP cloud to Router6 and Router7 that are shown in Figure 4.2 are loaded at 20%, 50%, and 80%. The simulations use three types of traffic types; Hyper Text Transfer Protocol (HTTP), File Transfer Protocol (FTP), and Voice over Internet Protocol (VoIP).

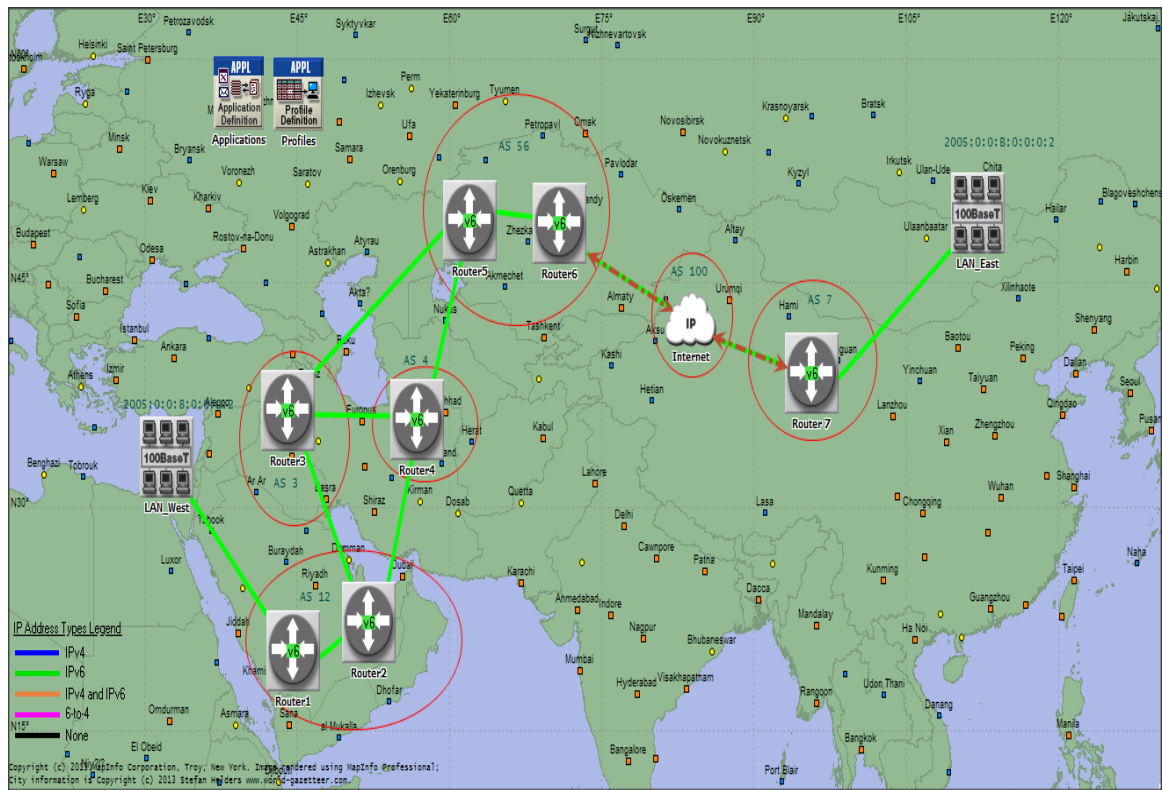


Figure 4.2 Network Showing the Links That will be Loaded With Traffic.

In this chapter we are looking at different outcomes. The first outcome is the convergence time of BGP protocol for the different solutions in the whole network. The

second outcome is the percentage of packet drop to the total number of application packets. The third outcome is the throughput in the link between Router2 and Router3 and the link between Router2 and Router4 in both directions. Also, we are looking for the application traffic sent from and received by LAN_East as well as the response time for an HTTP page and an FTP download. Each simulation runs for 2000 seconds during which the blackholing starts at 300 seconds, and the solution is applied at time 350. The HTTP and FTP simulations run for 20 times. While, the VoIP simulation is run for only 5 times due to the long time it takes to run the simulation. The mean and the confidence interval are displayed in some figures while in some figures we display only the mean, as to keep the figures legible. The VoIP experiments of the link load of 50% and 80% with an Internet delay of 5 seconds are not included due to several problems encountered in the simulation, as it will be explained in section 4.4.

4.2. Simulation Results and Analysis

This section is divided into three subsections. The first subsection shows the percentage of traffic drop results. The second subsection illustrates the convergence time results. The third subsection illustrates the performance figures of the throughput results.

4.2.1. Percentage of traffic drop

Figure 4.3 shows the dropped packets percentage for each simulation. The x-axis presents the different simulations configurations in terms of application type, the mean of the exponentially distributed delay of the Internet, and the link load. On the other hand, the y-axis shows the percentage of packet drop. The vertical bars in Figure

4.3 are the confidence intervals of the readings with 95% confidence interval.

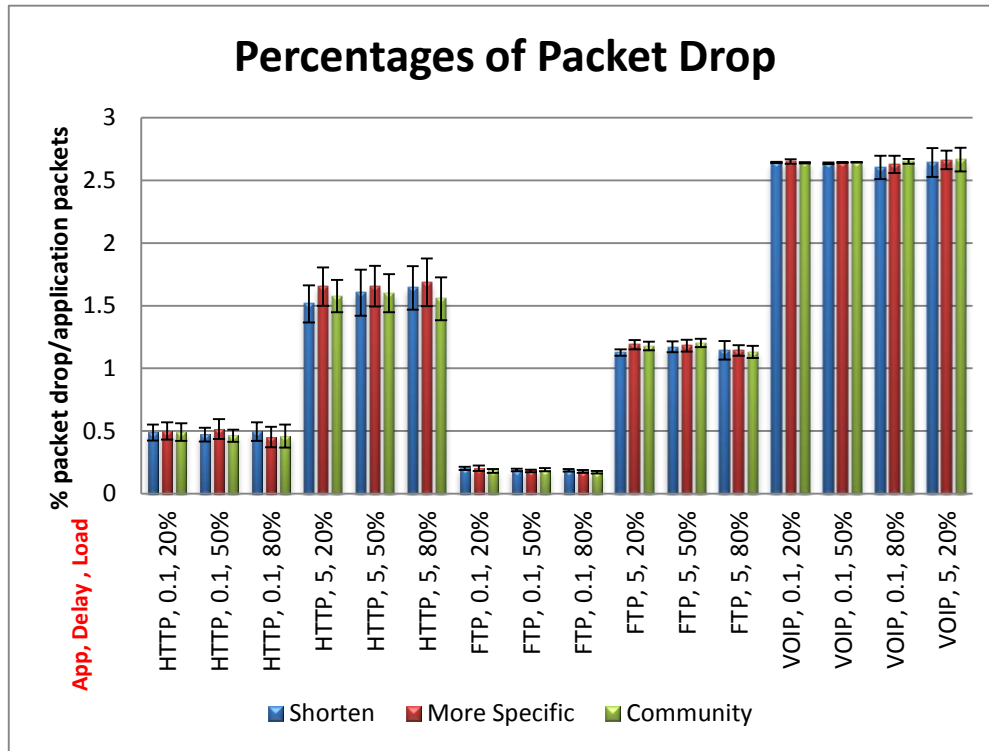


Figure 4.3 Packet Drop Percentages.

It is clear from Figure 4.3 that the FTP and the HTTP applications have lower dropped packets percentage than the VoIP application. On the other hand, the VoIP application with a 0.1 second Internet delay has the highest dropped packets percentage and it is about 5 times larger than the dropped packets percentage of HTTP and FTP. The reason for the high dropped packets percentage in VoIP is related to the nature of the VoIP traffic as it is a real time application which runs over UDP with a traffic rate that remains the same even during the blackholing period. The percentage of packet drop in HTTP is about double the percentage of packet drop in FTP application when 0.1 second Internet delay is used while the HTTP packet drop is slightly higher than that

of FTP when 5 seconds Internet delay is used. The difference in dropped packet percentage between HTTP and FTP is due to the protocol behavior of each protocol. And how each protocol is configured for further explanation can be found in Alrefai [1]. Moreover, the packet drop for both HTTP and FTP with Internet delay of 5 seconds is three times larger than for the case of 0.1 second Internet delay due to increasing the delay of the Internet. Figure 4.3 clearly shows that there is no significant effect of using different loads in the percentage of packet drop when the Internet delay is 0.1 second. However, some results show little increase in packet drop with different load, such as for HTTP application, when the delay of the Internet is 5 seconds. As for the confidence interval, Figure 43 shows that as the Internet delay increase the confidence interval increases due to the increased amount of randomness of the delay that the packets will experience.

4.2.2. Convergence Time

Figure 4.4 shows the convergence time for the 0.1 seconds Internet delay as a mean delay of each of the BGP-based solutions with different traffic loads.

The y-axis represents the convergence time in seconds and the x-axis displays the experiment with different traffic loads. As shown in Figure 4.4, the effect of the traffic load on the convergence time is very small. In general, the More Specific Prefix approach has the lowest convergence time even with the different traffic loads. This is because the approach needs to advertise less prefixes than in the shortening and the community approaches.

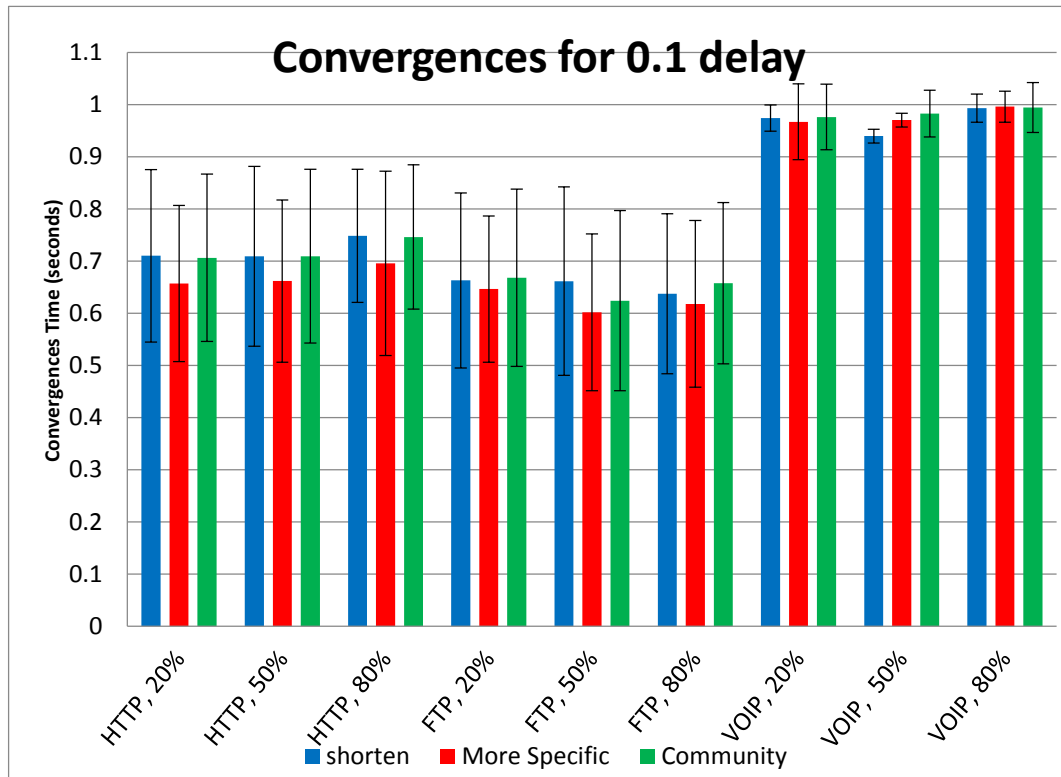
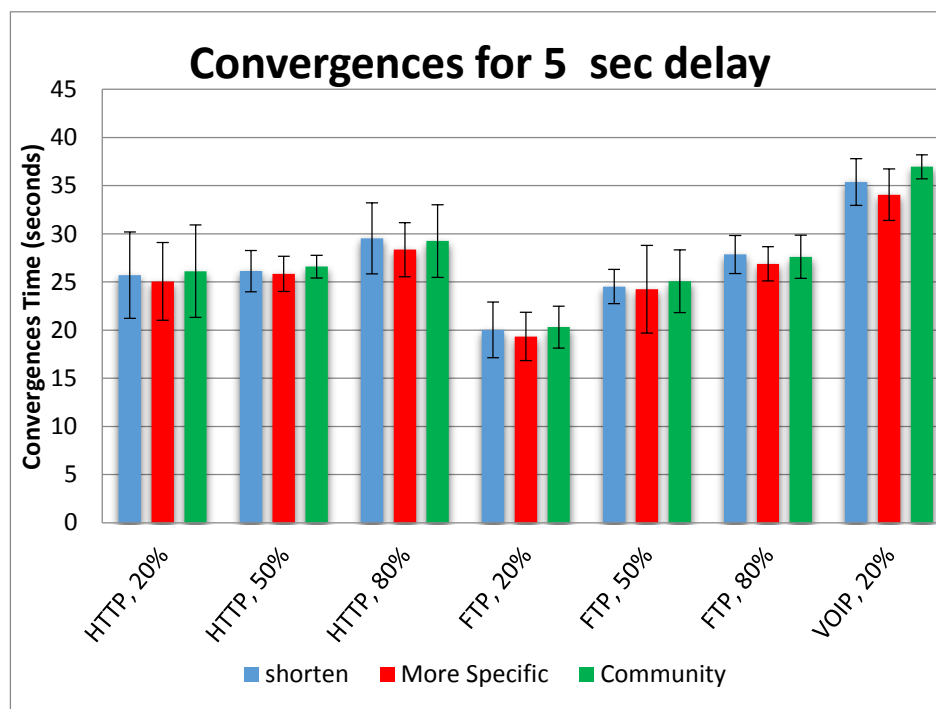


Figure 4.4 BGP Convergence Time for 0.1 Delay of Internet.

As shown in Figure 4.4 there is not much effect of loading the two links have on the convergence time. However, VoIP has higher convergence time when compared to HTTP and FTP. The increase in the convergence time in VoIP happens due to the high traffic demand which causes a delay in delivering BGP messages. On the other hand, HTTP has a slight increase in convergence time when compared against FTP because of the higher number of packets sent by HTTP as a result of having a shorter interarrival time than for FTP.



[Figure 4.5 BGP convergence time for 5 second delay.]

Figure 4.5, shows the convergence time for the 5 seconds Internet delay as a mean delay of each of the BGP-based solutions with different traffic loads. The y-axis represents the convergence time in seconds and the x-axis displays the experiment with different traffic loads. The effect of increasing the Internet delay on convergence is clear when compared to the convergence time for the 0.1 seconds delay of the Internet.

Figure 4.5 shows that the More Specific Prefix approach has the lowest convergence time even with the different traffic loads. VoIP has higher convergence time when compared to HTTP and FTP. The increase in the convergence time in VoIP happens due to the high traffic demand which causes some delay in delivering the BGP messages. The loading of the two links has no significant effect on the convergence time in general while in the case of FTP the higher the link-load the higher the convergence time.

4.2.3. Throughput

In this section, we discuss the throughput, measured in bits per second, in both directions of the links connecting the local router to the malicious and the non-malicious routers. The throughput between Router2 and Router3 shows the effect of blackholing. Moreover, the throughput between Router2 and Router3 shows the effect of using different solutions. In the following figures the mean throughput of multiple runs is shown. Note that the baseline simulation results when there is no malicious activity is provided in Appendix B.

Figure 4.6 shows the throughput, in bits per second, for HTTP traffic from Router2 to Router3.

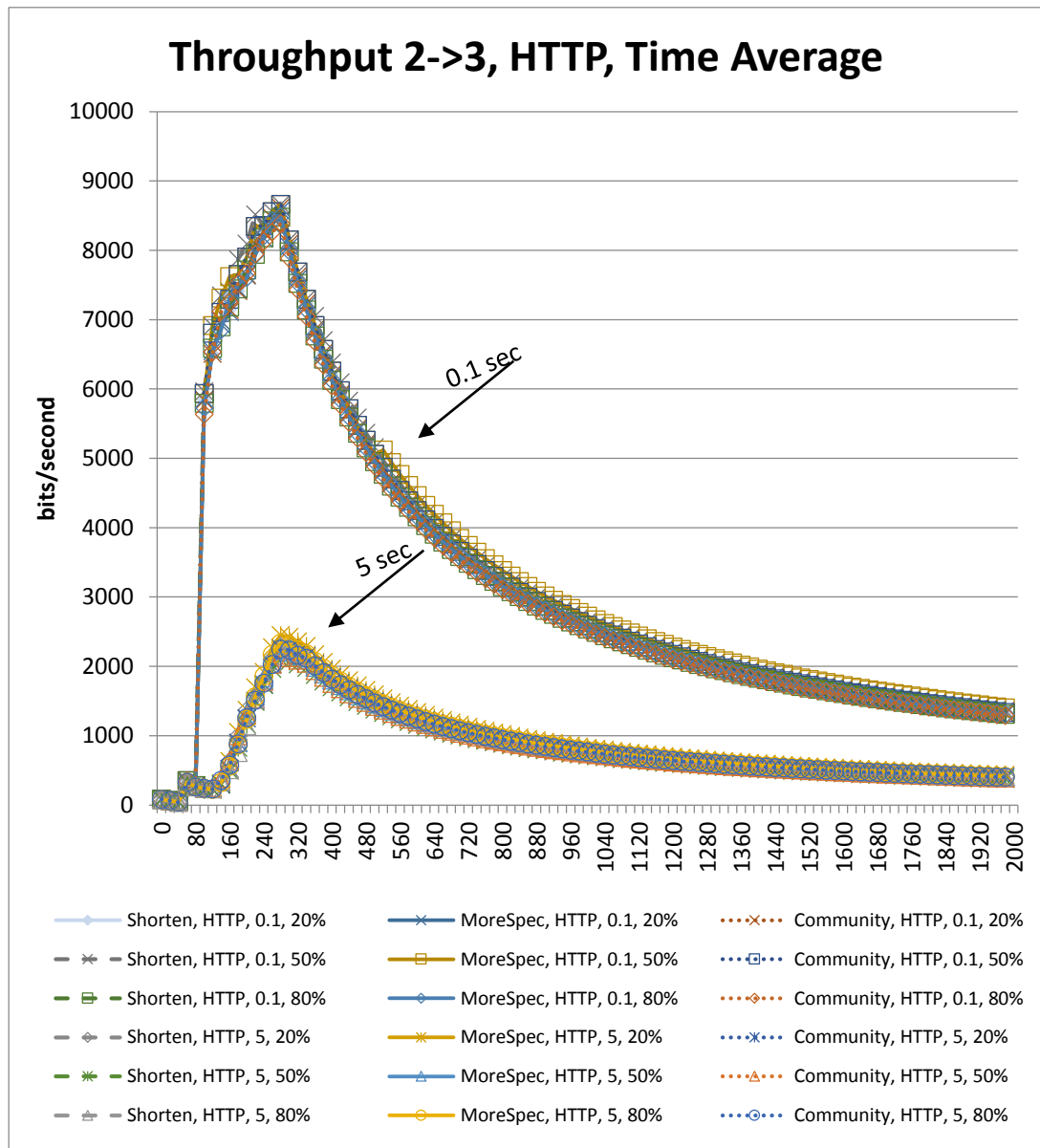


Figure 4.6 Outgoing Throughput From Router2 to Router3 for HTTP.

Figure 4.7 shows the throughput, in bits per second, for FTP traffic from Router2 to Router3. Note that the large throughput that is observed around 80 second is mostly due to the number of TCP connections of that FTP establish at the beginning of the FTP session.

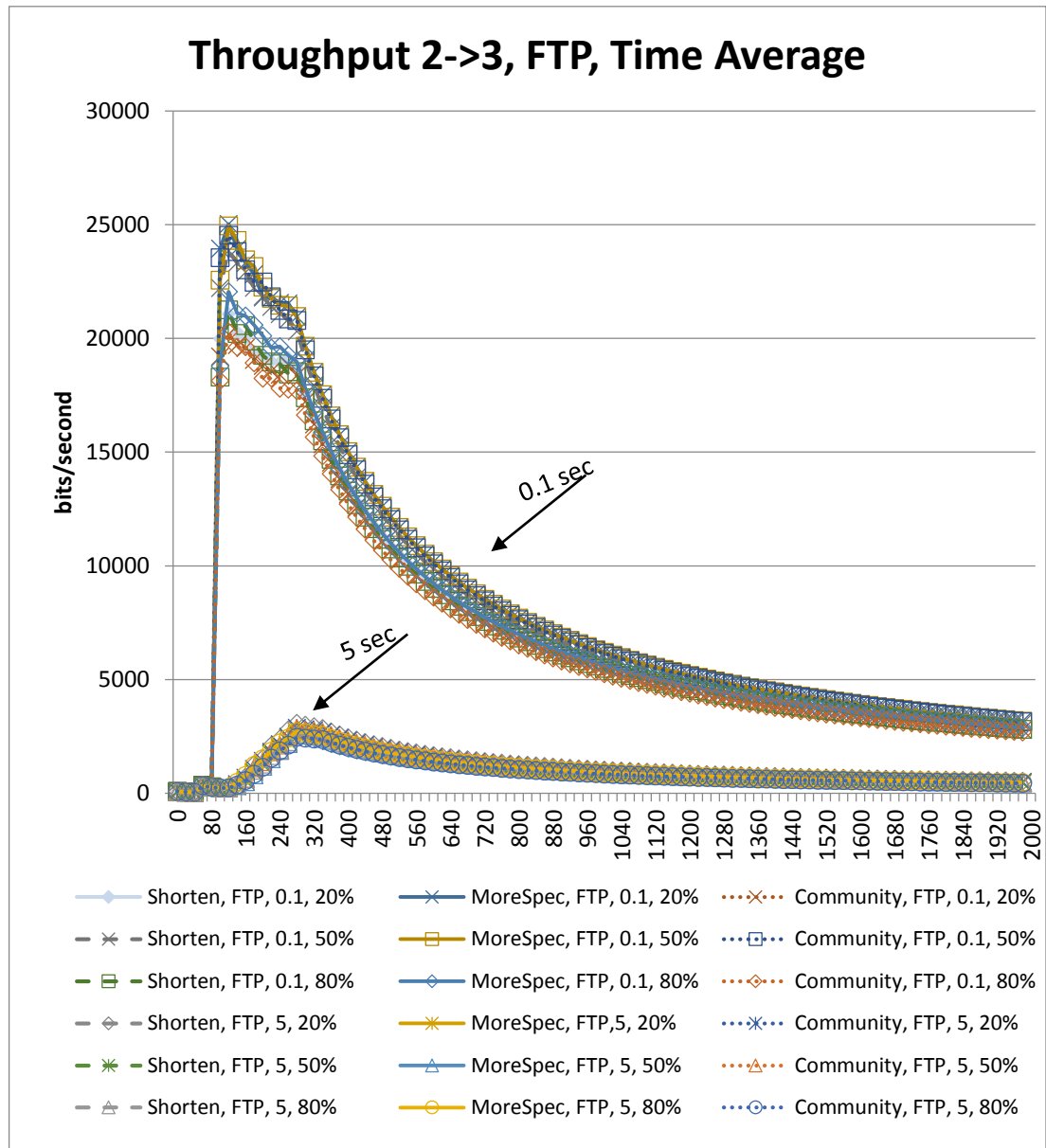


Figure 4.7 Throughput From Router2 to Router3 for FTP Application.

Figure 4.8 shows the throughput, in bits per second, for VoIP traffic from Router2 to Router3.

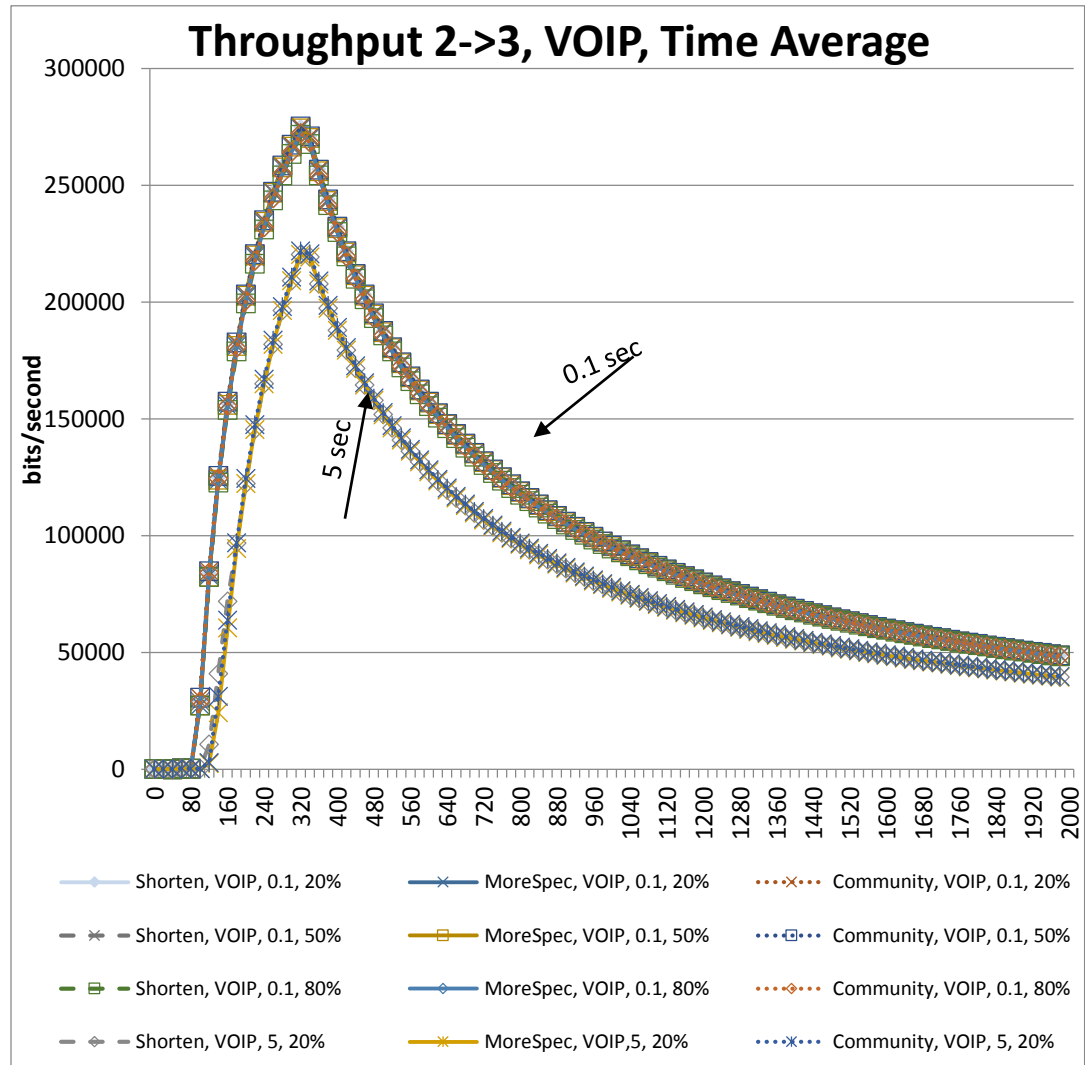


Figure 4.8 Throughput From Router2 to Router3 for VoIP Application.

As shown in Figure 4.6, 4.7, and Figure 4.8, for each of the three applications all solutions result in almost the same throughput in terms of different link loads. On the other hand, for each application all solutions produce different throughput for different Internet delays. The increase of the Internet delay causes a decrease in the throughput. The reason behind such a decrease is attributed to the fact that an increase in the Internet delay triggers the TCP congestion control which causes a decrease in the transmission rate. On the other hand, the VoIP throughput is less impact by the Internet delay because

VoIP is a real time protocol that operate on top of the UDP protocol which has no congestion control.

In addition, we notice from Figures 4.6, 4.7, and 4.8 we can notice that HTTP has the lowest throughput and this mainly due to the HTTP configuration set in the simulation where the average page size that the clients request from the server is 7250 bytes while the size of the file that FTP download, for example, is 50000 bytes.

The following figures show the throughput for different applications from Router3 to Route2. Figure 4.9 shows the throughput, in bits per second, for HTTP traffic from Router3 to Router2.

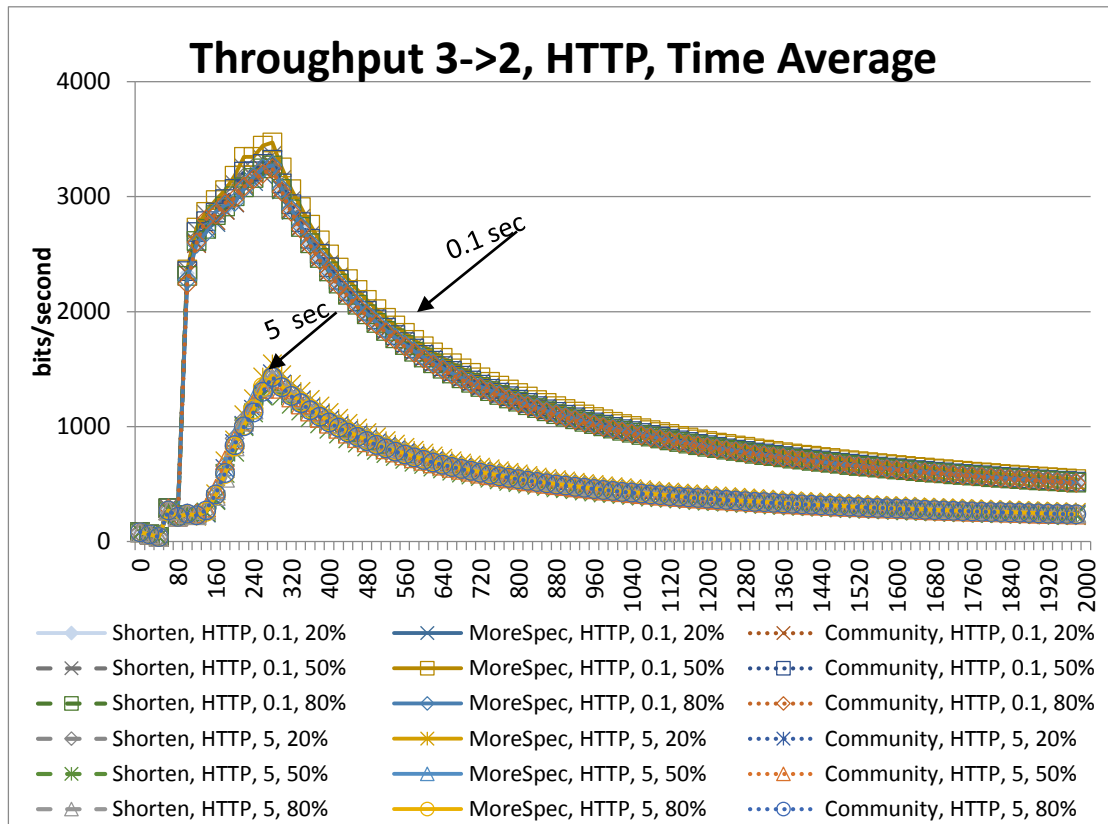


Figure 4.9 Incoming Throughput to Router2 from Router3 for HTTP Application.

Figure 4.10 shows the throughput, in bits per second, for FTP traffic from Router3 to Router2. From Figure 4.9 and Figure 4.10 it can be noticed that the higher throughput takes place when the Internet delay is less.

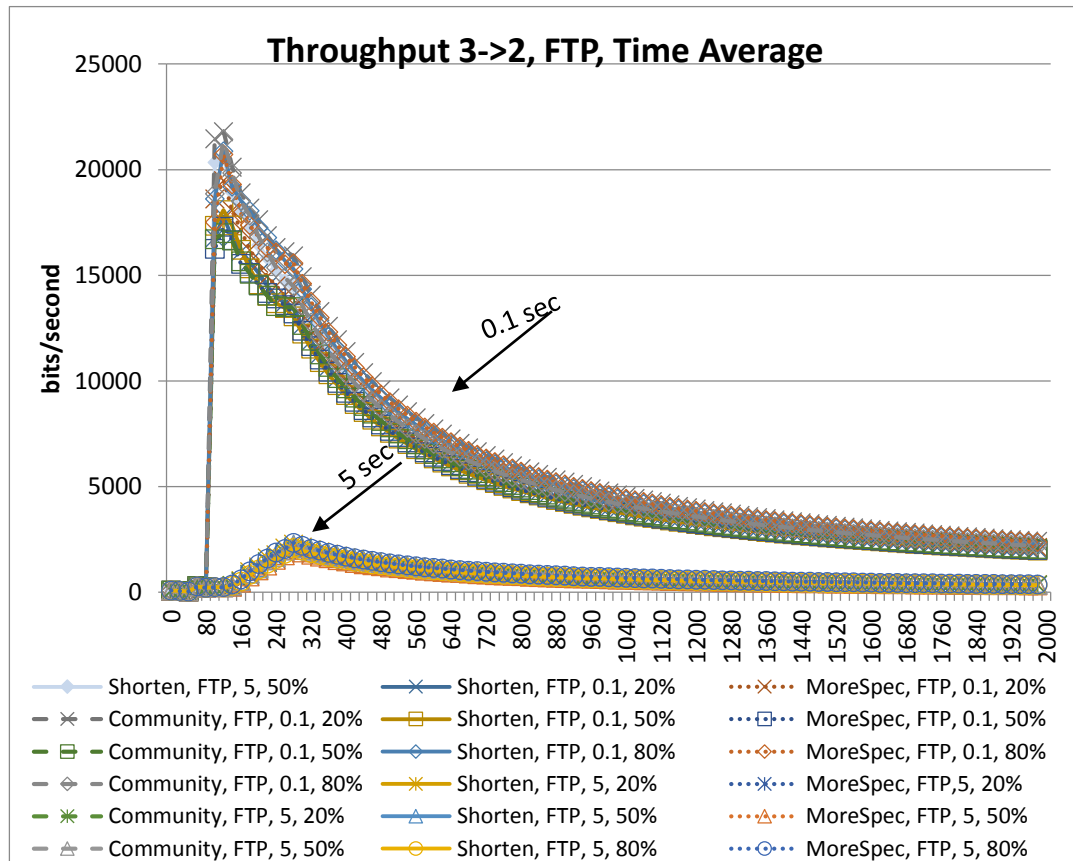


Figure 4.10 Throughput from Router2 to Router3 in FTP Application.

Figure 4.11 shows the throughput, in bits per second, for VoIP traffic from Router3 to Router2. From the figure it can be noticed that the higher throughput takes place when the Internet delay is less. As shown in 4.9, 4.10, and 4.11 figures that the load has no impact on VoIP, HTTP or FTP.

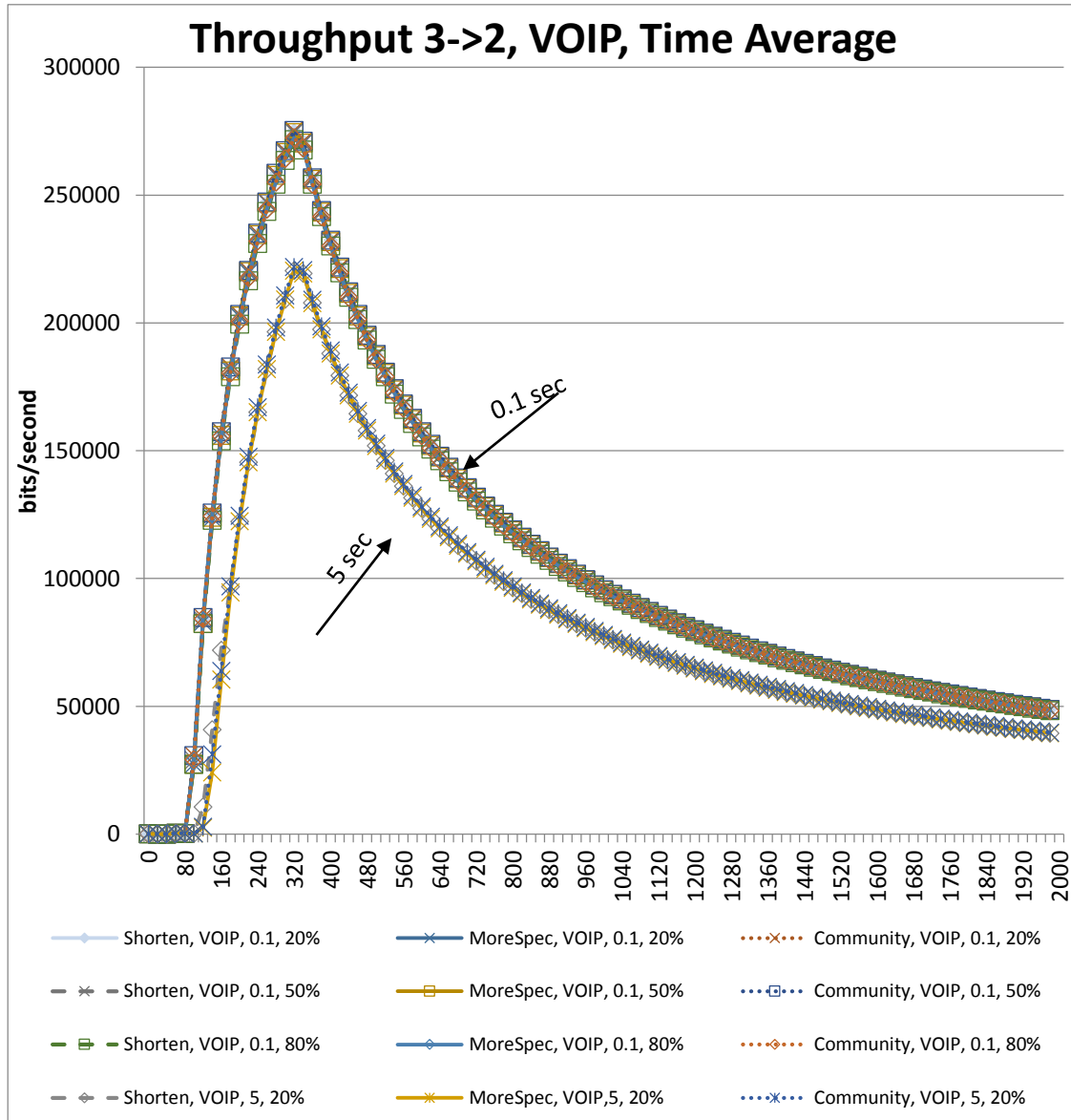


Figure 4.11 Throughput from Router2 to Router3 for VOIP application.

The previous figures illustrated the throughput between Router2 and Router3 in both directions. They show that throughput starts to decrease at time 300 due to the start of blackholing at that time. Note that the solutions start at time 350.

Note that the throughput does not reach 0 bits per second because the figures reflect a time average throughput; and therefore earlier throughput values prevent the overall throughput from reaching 0 bits per second.

The following figures illustrate the effect of the applied solutions by studying the traffic between Router2 and Router4 in both directions. Figure 4.12 shows the throughput, in bits per second, for HTTP traffic from Router2 to Router4.

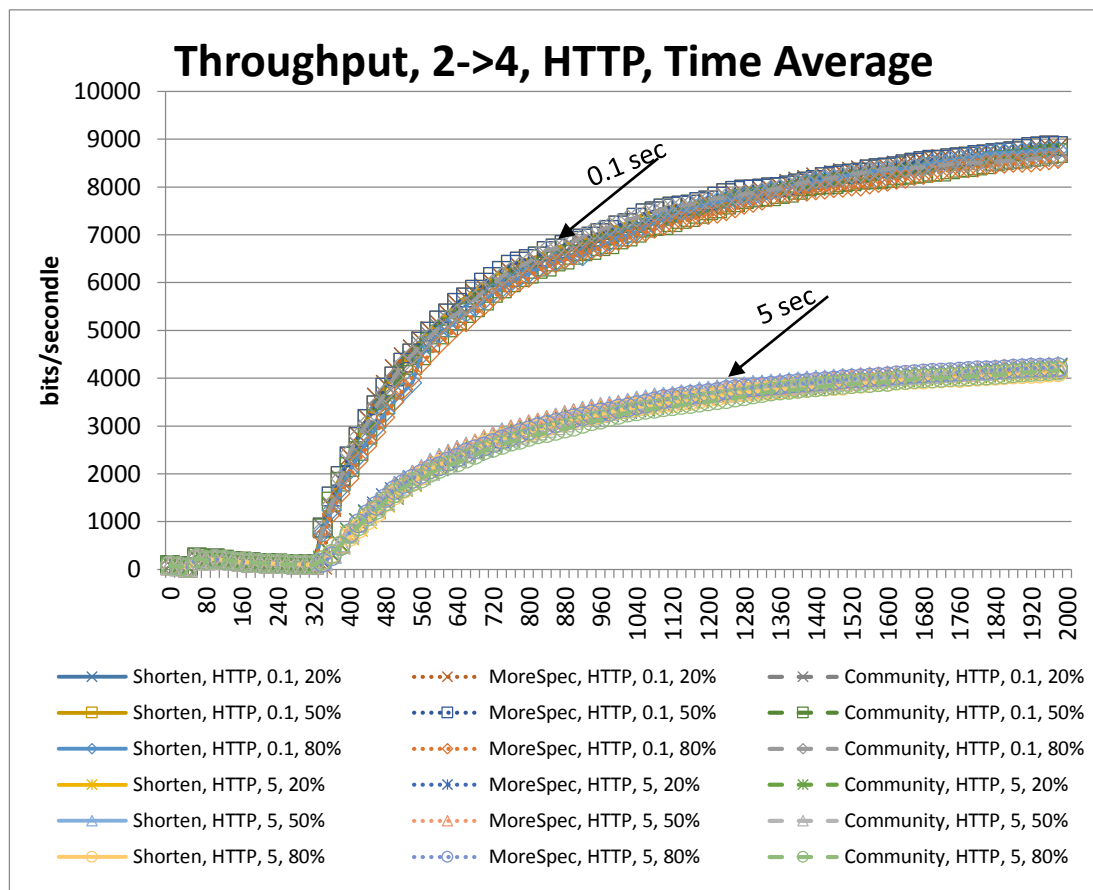


Figure 4.12 Outgoing Traffic From Router2 to Router4 for HTTP Application.

From the figure it is clear that the throughput for Internet delay of 5 seconds is about half the throughput of 0.1 second Internet delay. For the same Internet delay, different solutions show a similar behavior in terms of outgoing traffic towards Router4.

Figure 4.12 shows that the HTTP throughput reaches almost 9000 bits per second for the Internet delay 0.1 seconds which is the same throughput as in Figure 4.6. On the other hand, the HTTP throughput for the Internet delay 5 seconds in Figure 4.12 is over 4000 bits per second while in Figure 4.6 it is slightly above 2000 bits per second. Because the 5 seconds Internet delay causes a high initial convergence time, the throughput in Figure 4.6 does not reach the same level as in Figure 4.12 due to the triggering of the blackholing.

Figure 4.13 shows the throughput, in bits per second, for FTP traffic from Router2 to Router4. In the figure, no difference can be seen when using 0.1 seconds or 5 seconds Internet delay due to configuring FTP with an inter-request time of 360 seconds which less FTP requests and causing the effect of the Internet delay to diminish. Note that when the inter-request time is set to 60 seconds there will be a difference between the 0.1 seconds delay and the 5 seconds delay as shown in Appendix D.

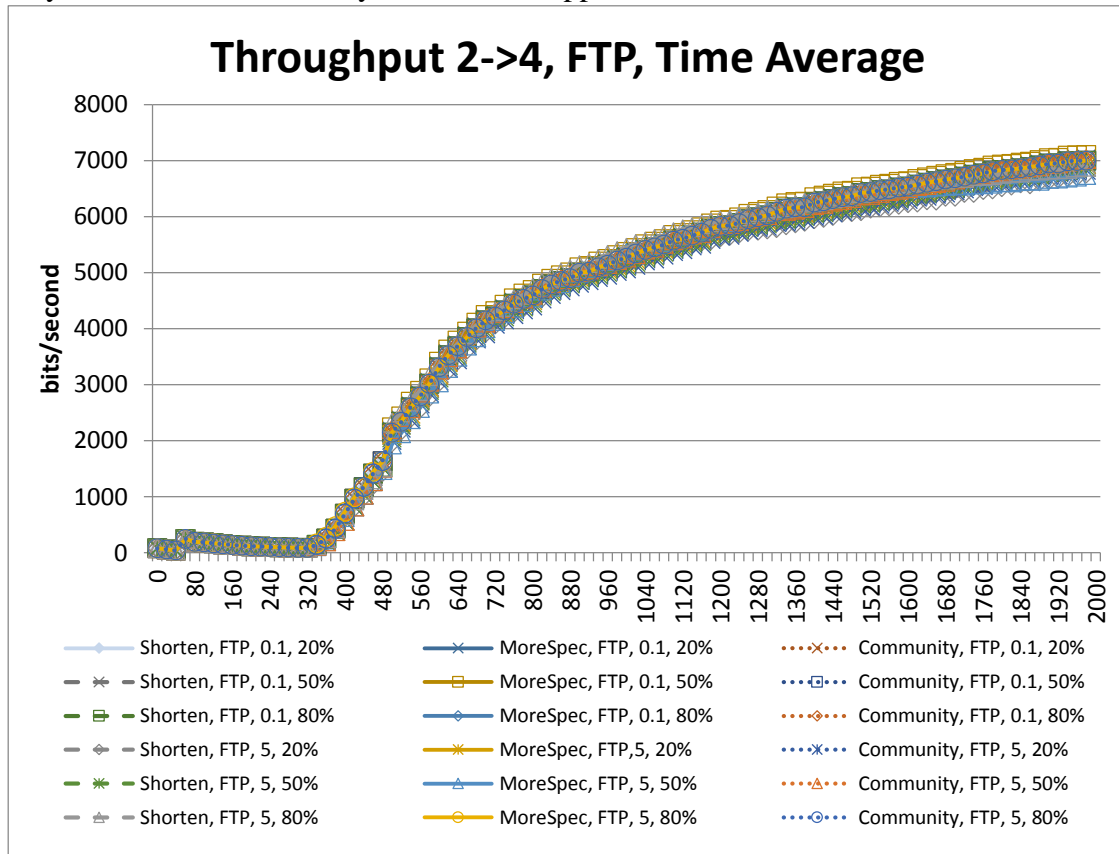


Figure 4.13 Throughput from Router 2 to Router 4 for FTP Application.

Figure 4.13 shows clearly a difference in the throughput between the 0.1 and the 5 seconds Internet delay when comparing Figure 4.7 to Figure 4.13. The reason for this difference is the slow convergence of the 5 seconds delay.

Figure 4.14 shows the throughput, in bits per second, for VoIP traffic from Router2 to Router4. VoIP is a real time application which runs over UDP which does not have flow control nor does it have congestion control. Thus, we cannot notice any difference in the throughput of VoIP traffic for 0.1 seconds or 5 seconds Internet delay.

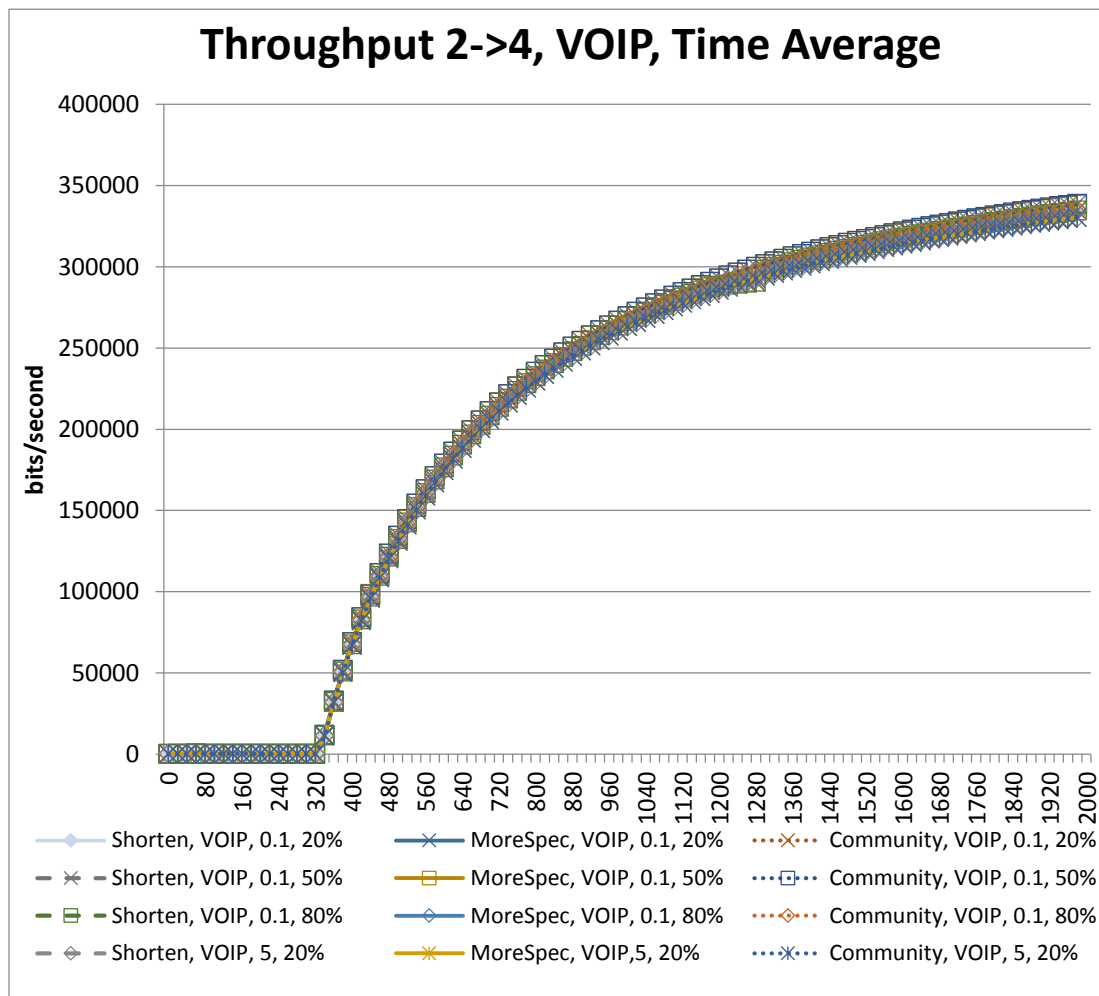


Figure 4.14 Throughput from Router2 to Router4 in VOIP Application.

Figure 4.15 shows the throughput from Router4 to Router2 for the HTTP application.

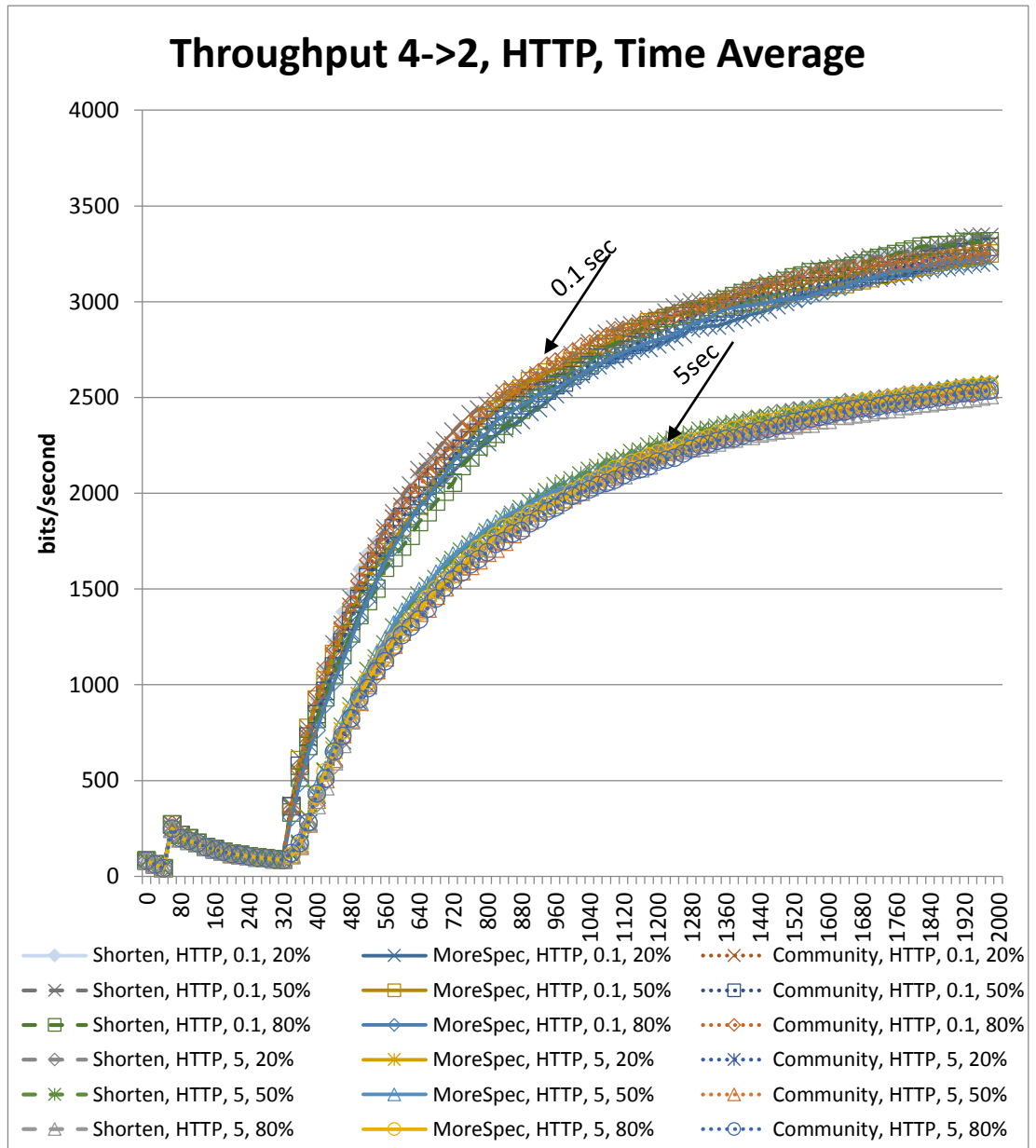


Figure 4.15 Throughput from Router4 to Router2 for HTTP Application.

Figure 4.15 shows a higher throughput for the 5 seconds Internet delay than the throughput of the 5 seconds Internet delay scenario shown in Figure 4.9. The lower

throughput noticed in Figure 4.9 is due to the fact that increase in the throughput was interrupted by the blackhoaling activity.

Figure 4.16 shows the throughput, in bits per second, for FTP traffic from Router4 to Router2. It is clear from the figure that a higher load results in slightly lower throughput in FTP. The reason behind this is that the FTP traffic is not exposed to TCP congestion control due to the inter-request time begin set to 360 seconds. Accordingly, there is no difference in the FTP throughput for different loads and different Internet delays.

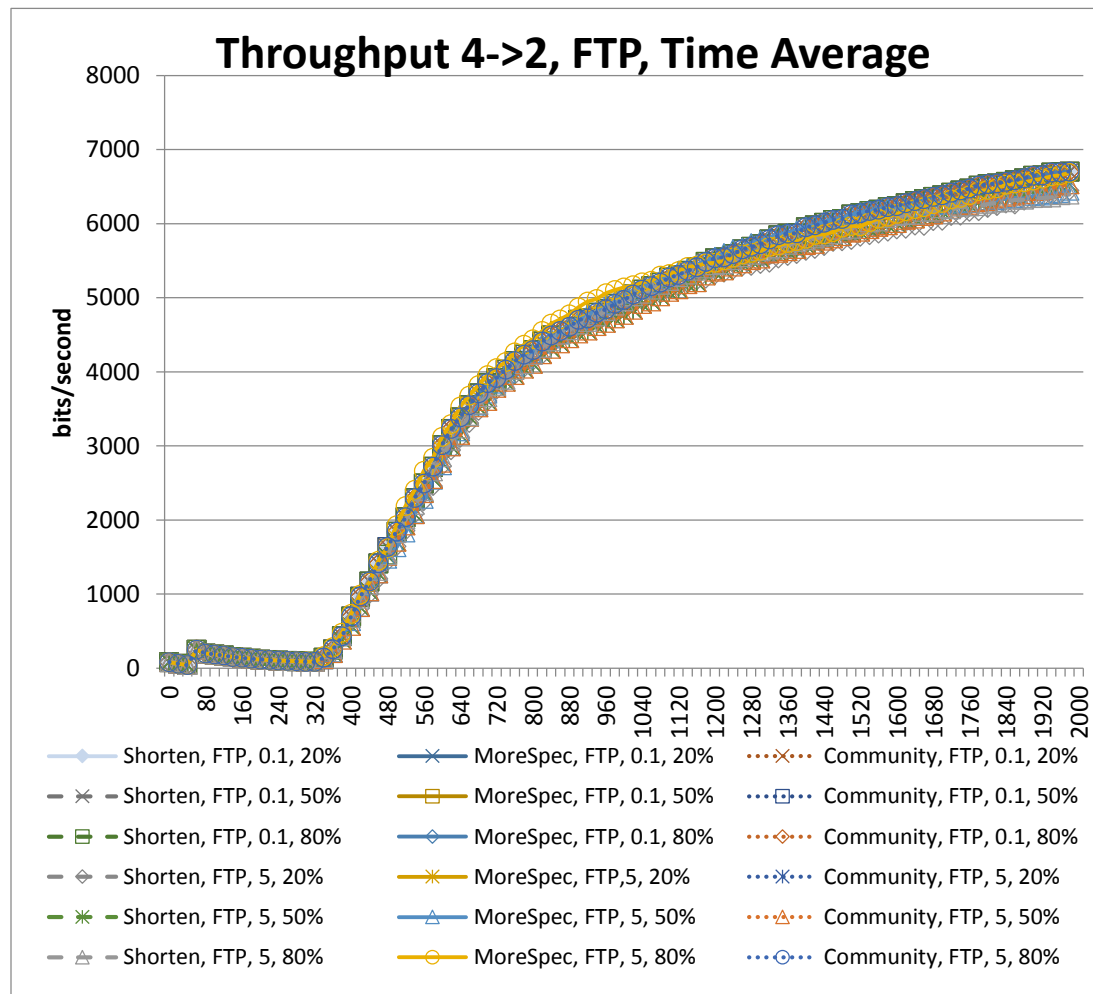


Figure 4.16 Throughput from Router4 to Router2 in FTP application.

From Figure 4.16 and Figure 4.13 the throughput does not change when changing either the load or the Internet delay for all the solutions due to higher inter-request time which prevents the TCP congestion control. Moreover, it is clear from the figures that the throughput is almost the same in both directions. The main reason for that is that the Command Mix for the FTP experiments is configured with the Command Mix set to 50%, which means that the 'Get' command is 50% and the 'Put' command is the other 50%. This results in similar traffic throughput. On the other hand, the HTTP throughput in Figure 4.15 is lower than the HTTP throughput in Figure 4.12 because the size of the file sent from the server to the client is larger than the requests sent from the client to the server. Hence, the amount of traffic from Router2 to Router4 is more than the amount of traffic from Router4 to Router2.

Figure 4.17 shows the throughput from Router4 to Router2 in the VoIP application. Increasing the load or the delay of the Internet does not have a significant impact on throughput for the same reasons as pointed out when discussing the results of Figure 4.14.

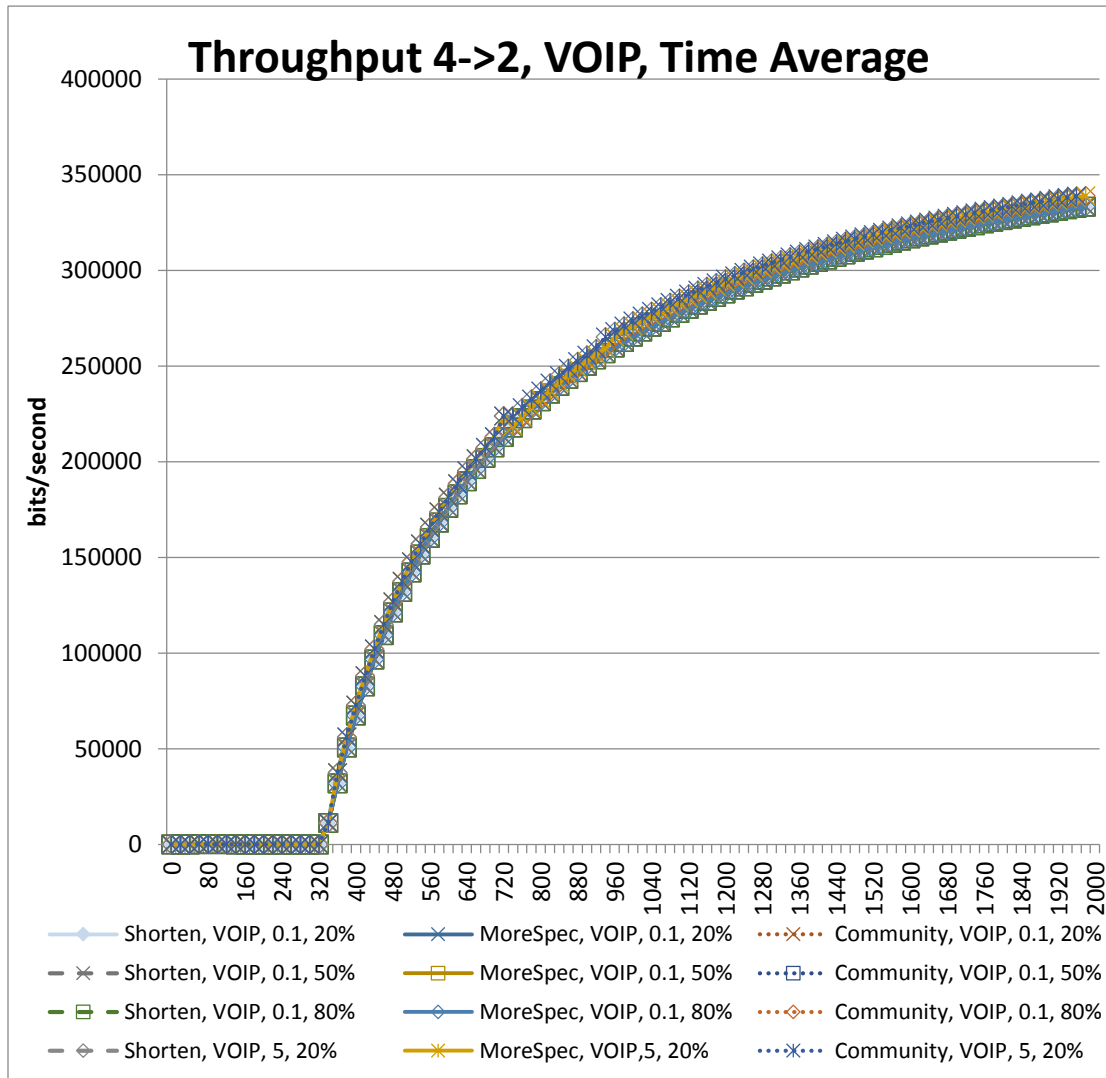


Figure 4.17 Throughput from Router4 to Router2 in VOIP Application.

4.2.3.1. Application Level Throughput

In this section, we discuss and present figures of the traffic sent and received by LAN_East for different applications.

Figure 4.18 shows the HTTP traffic sent from the LAN_East subnet.

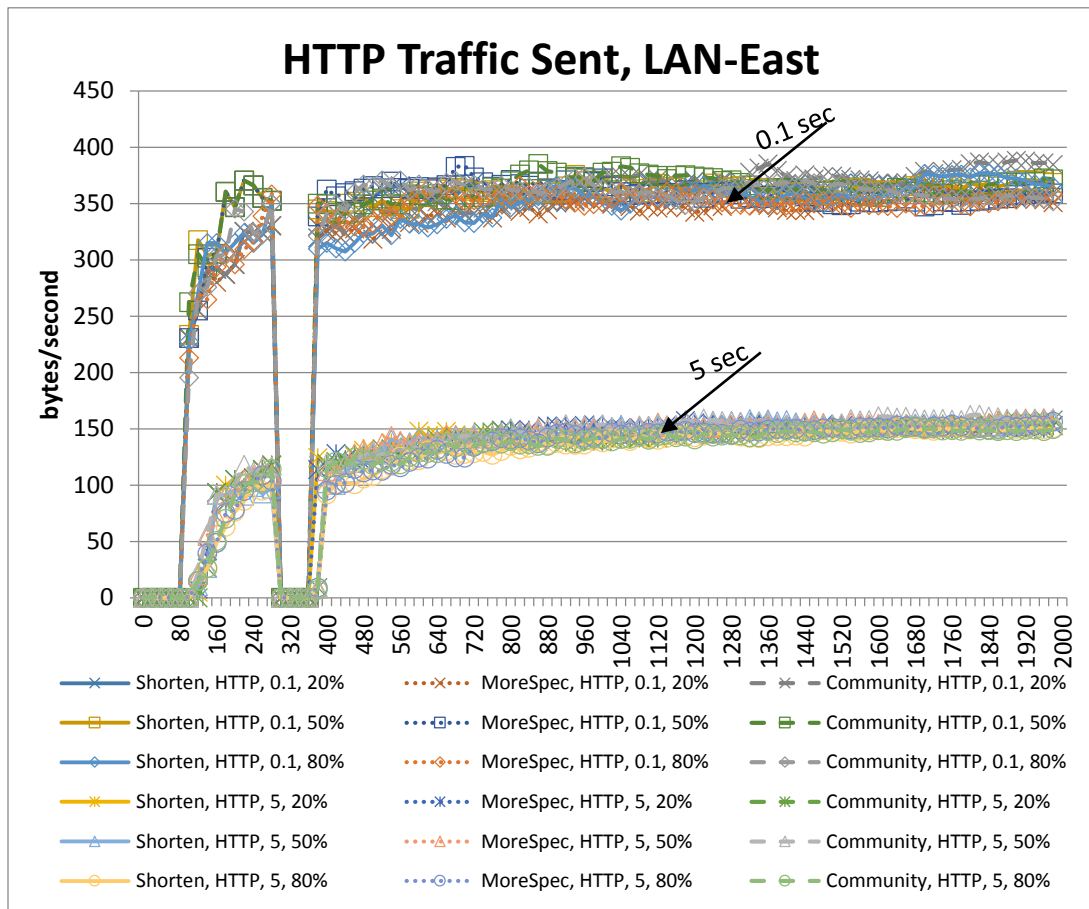


Figure 4.18 HTTP Packet Sent by LAN_East.

The figure also shows the effect of the Internet delay on the packets sent. The packets sent when the Internet delay is 0.1 second is almost double the packets sent with a 5 seconds Internet delay. Moreover, the figure shows that the traffic is zero during the time of blackholing (300-360).

Figure 4.19 shows the HTTP packets received by LAN_East.

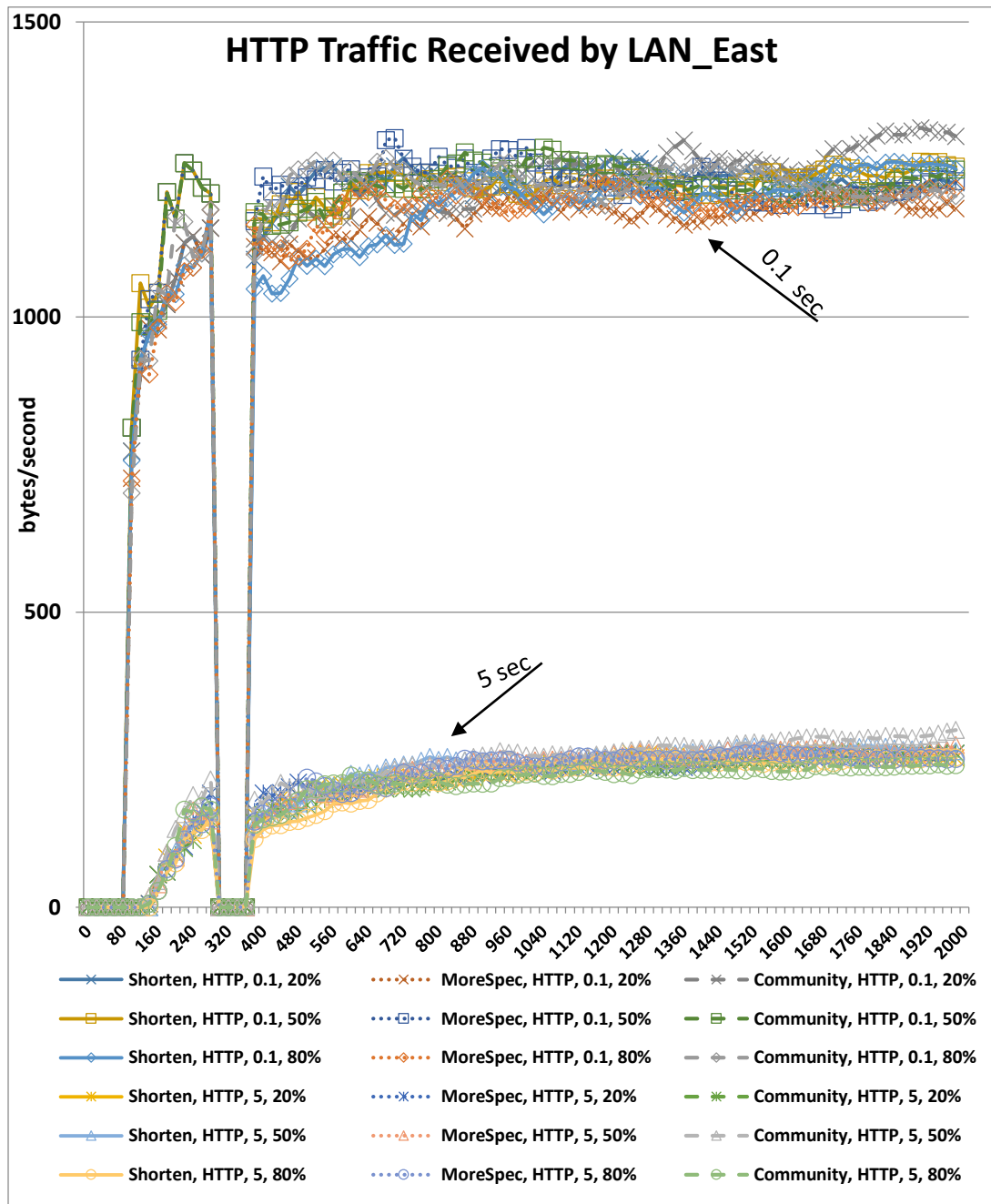
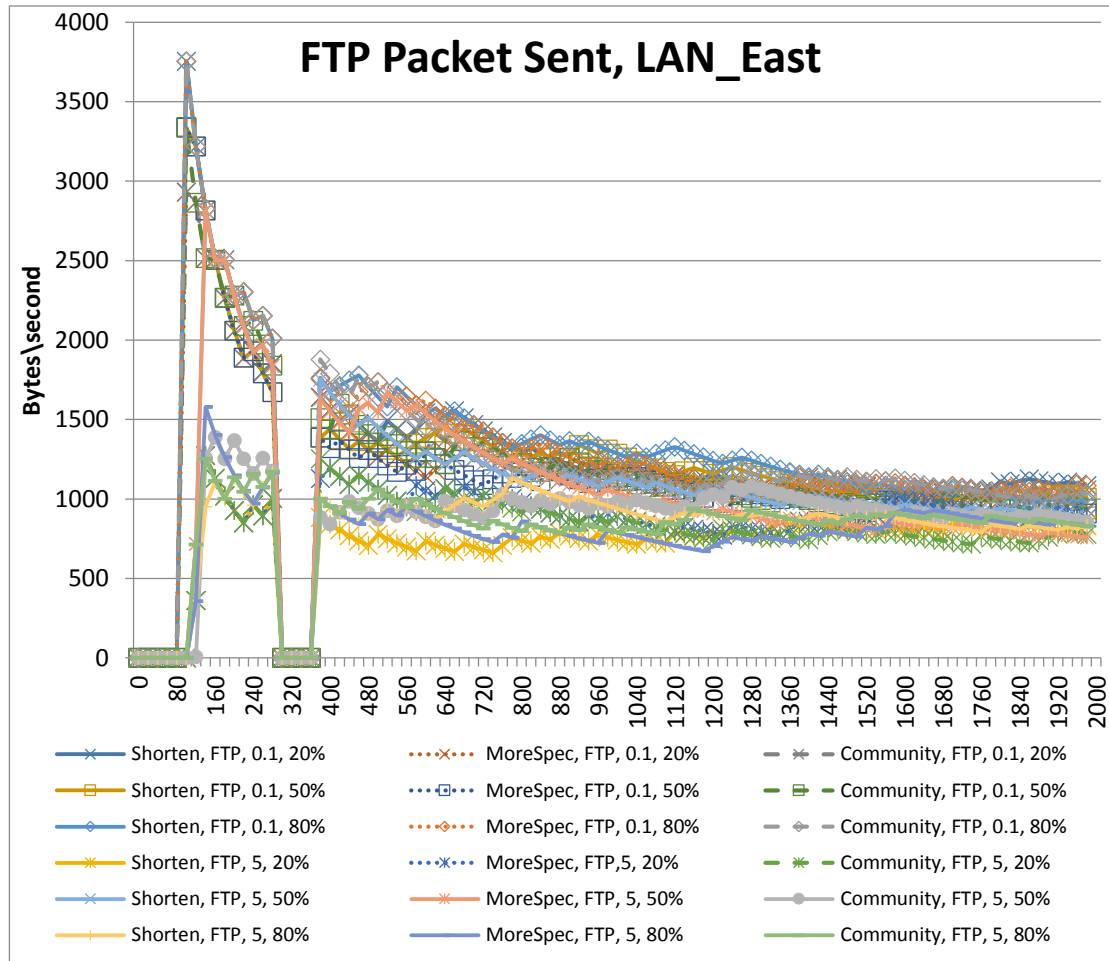


Figure 4.19 HTTP Packet Received by LAN_East.

The packets received reduce to 0 in all experiments during the period of blackholing. Figure 4.20 shows the FTP packets sent from the LAN_East subnet. The figure clearly shows that the number of packets sent during the time of blackholing reduces to zero.



[Figure 4.20 FTP Packets Sent from LAN_East.]

Figure 4.21 shows the FTP packets received by LAN_East. From the figure we can notice that there are no packets received during the period of blackholing. Moreover, there are no packets received for the 5 seconds Internet delay until the recovery from

blackholing. The main reason for this is the long inter-request time in conjunction with the higher convergence delay that is associated with the 5 seconds Internet delay. When the solution is applied and network converges, it starts the initialization of FTP which affects when LAN_East starts receiving packets from LAN_West.

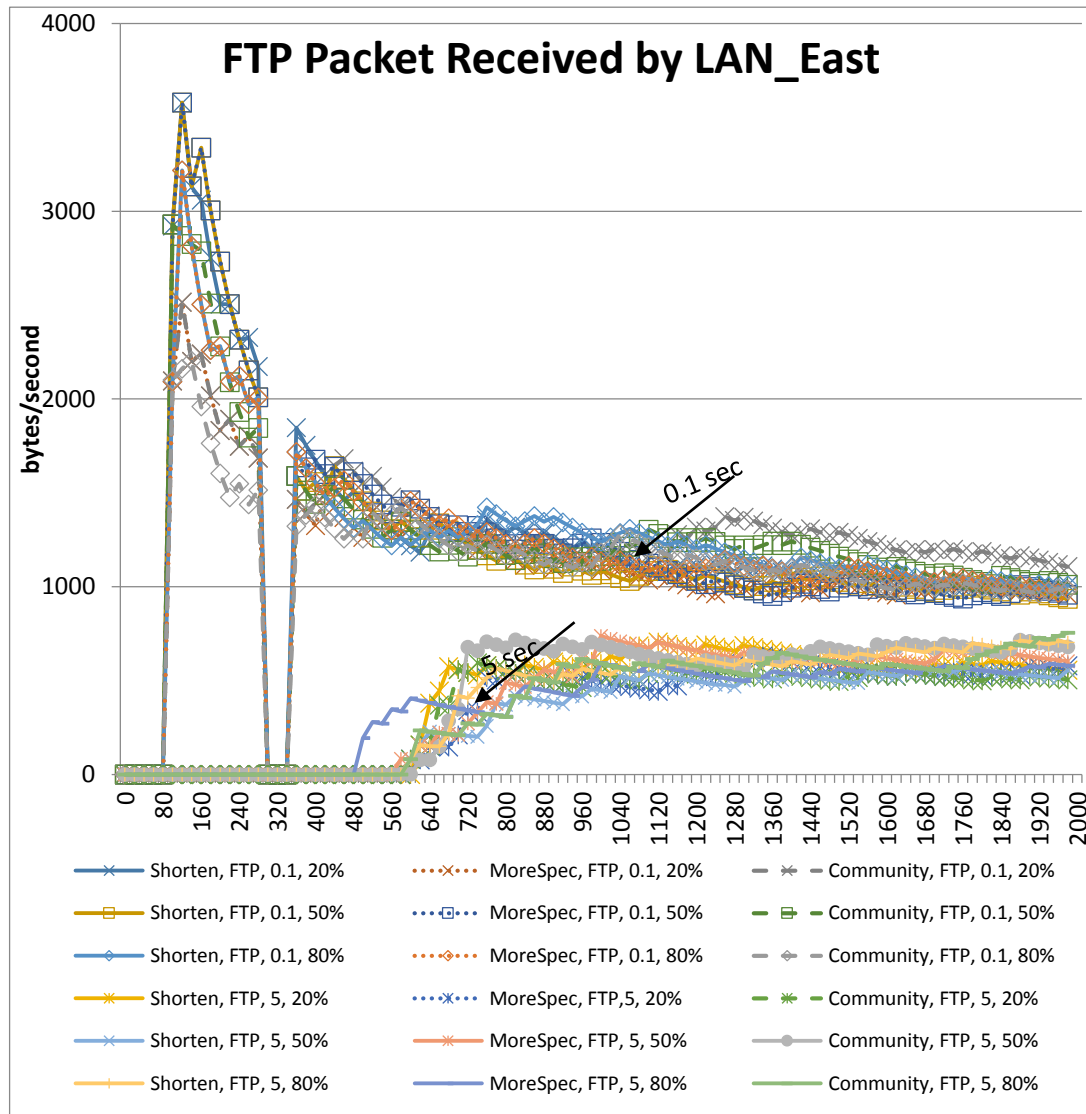


Figure 4.21 FTP Packet Received to LAN East.

Figure 4.22 shows the VoIP packet sent from LAN_East. The figure shows that the traffic load and the Internet delay have no effect on the amount of packets sent. The reason behind such a behavior is the fact that VoIP runs over UDP which is unaffected by the presence of blockholing.

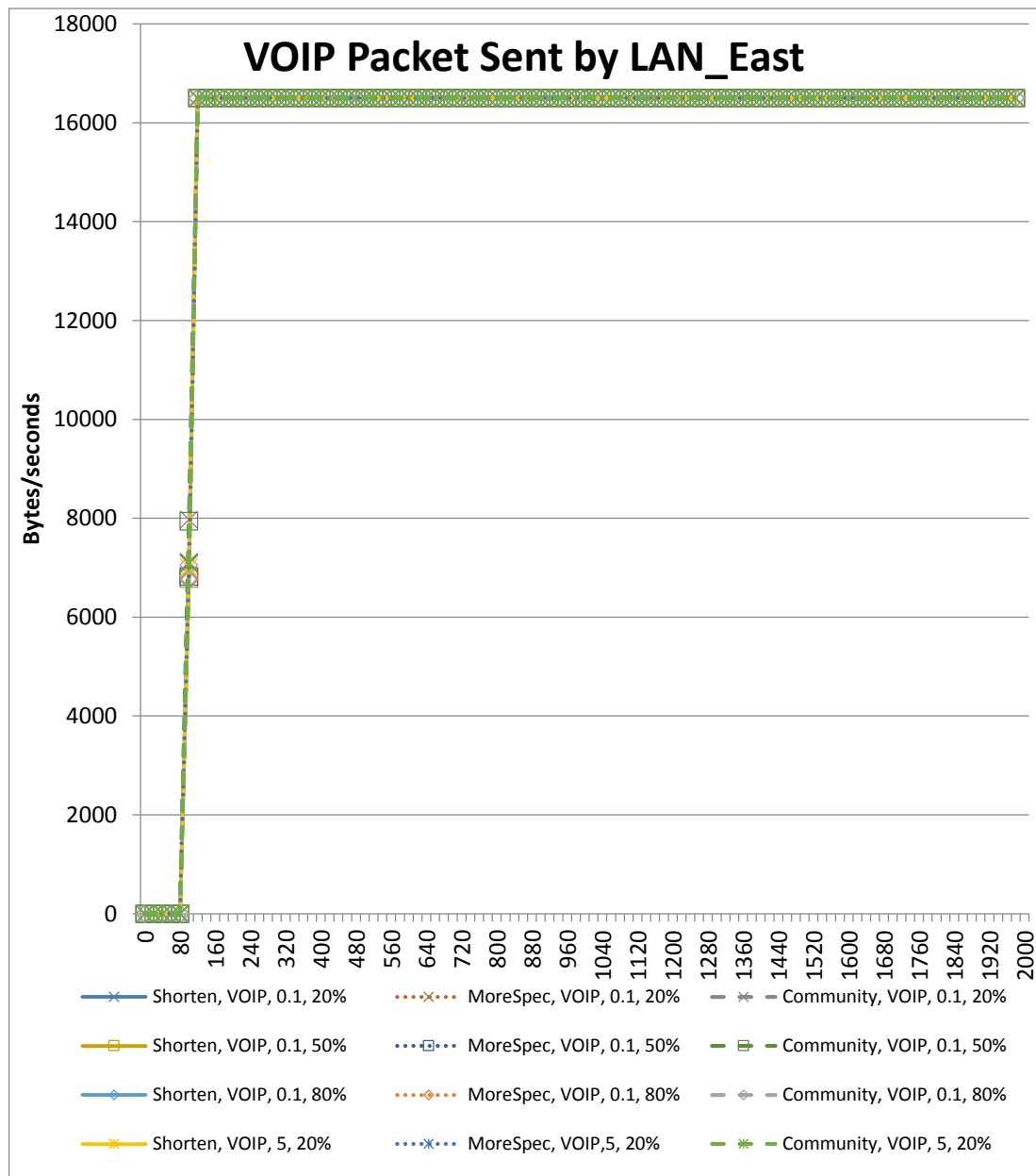


Figure 4.22 VoIP Traffic Sent from LAN_East.

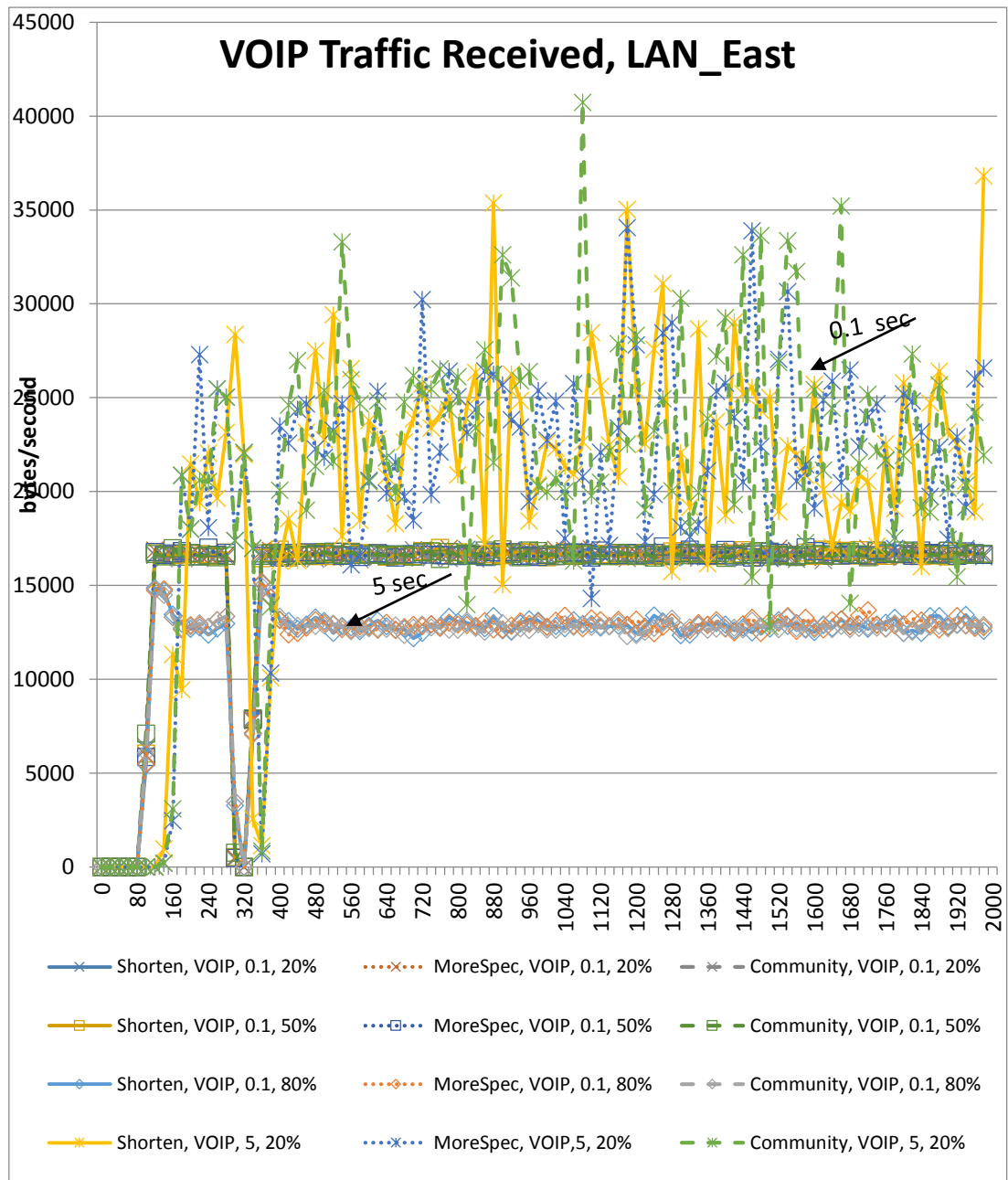


Figure 4.23 VoIP Traffic Received by LAN_East.

Figure 4.23 shows the VoIP packets received by LAN_East. Due to the exponential delay with 5 seconds as a mean for the VoIP application it can be clearly noticed the high traffic fluctuating for the 5 seconds Internet delay scenario. Also, the figure shows that the traffic is zero during the time of blackholing.

Figure 4.24 shows the page response time for the HTTP application for different solutions, load, and delay of the Internet.

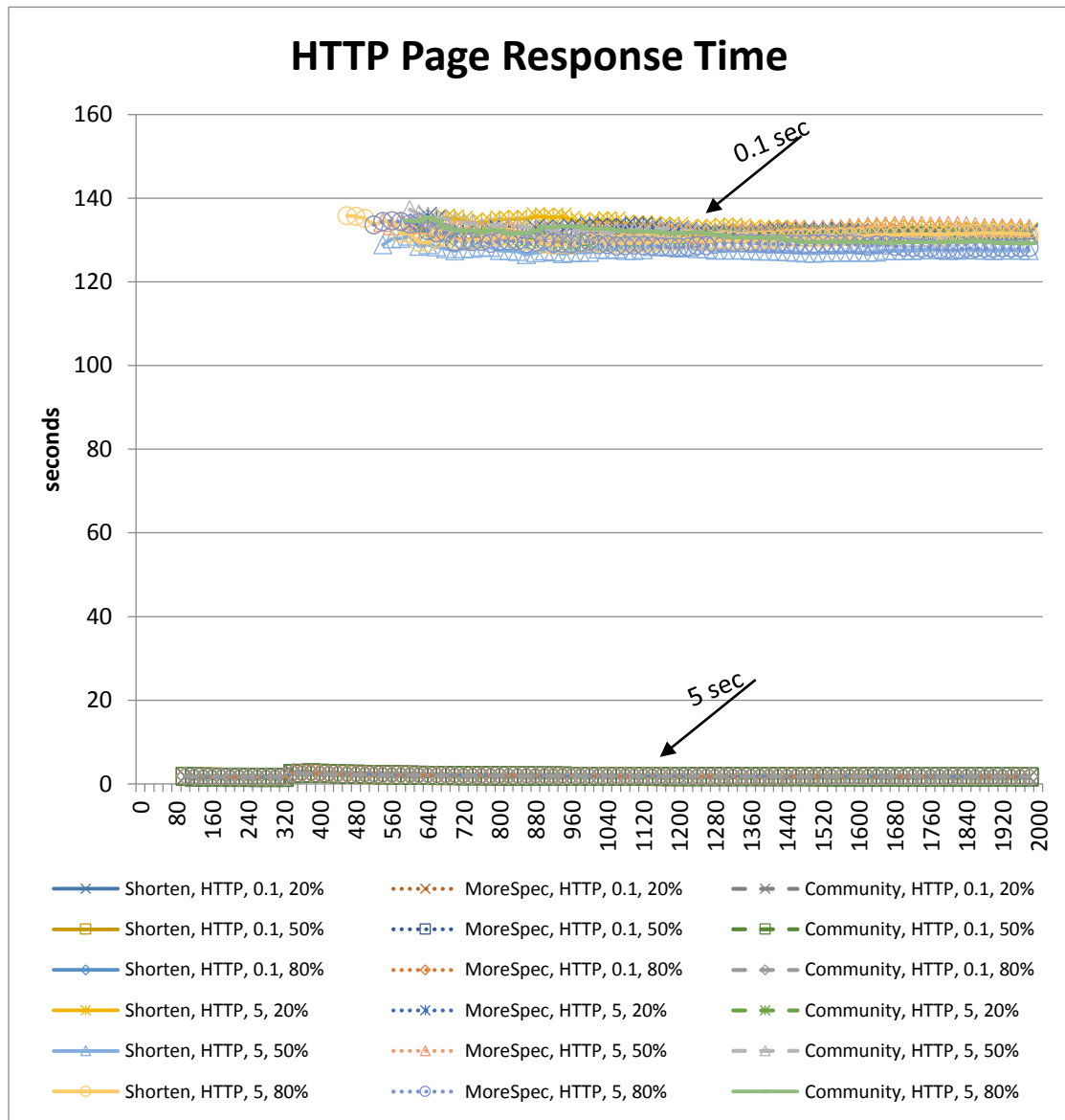


Figure 4.24 Page Response time for HTTP Client.

Figure 4.24 shows that the page response time is higher when the Internet delay is high. The FTP download response time shown in Figure 4.25 has the same characteristics as that shown in Figure 4.24 for HTTP.

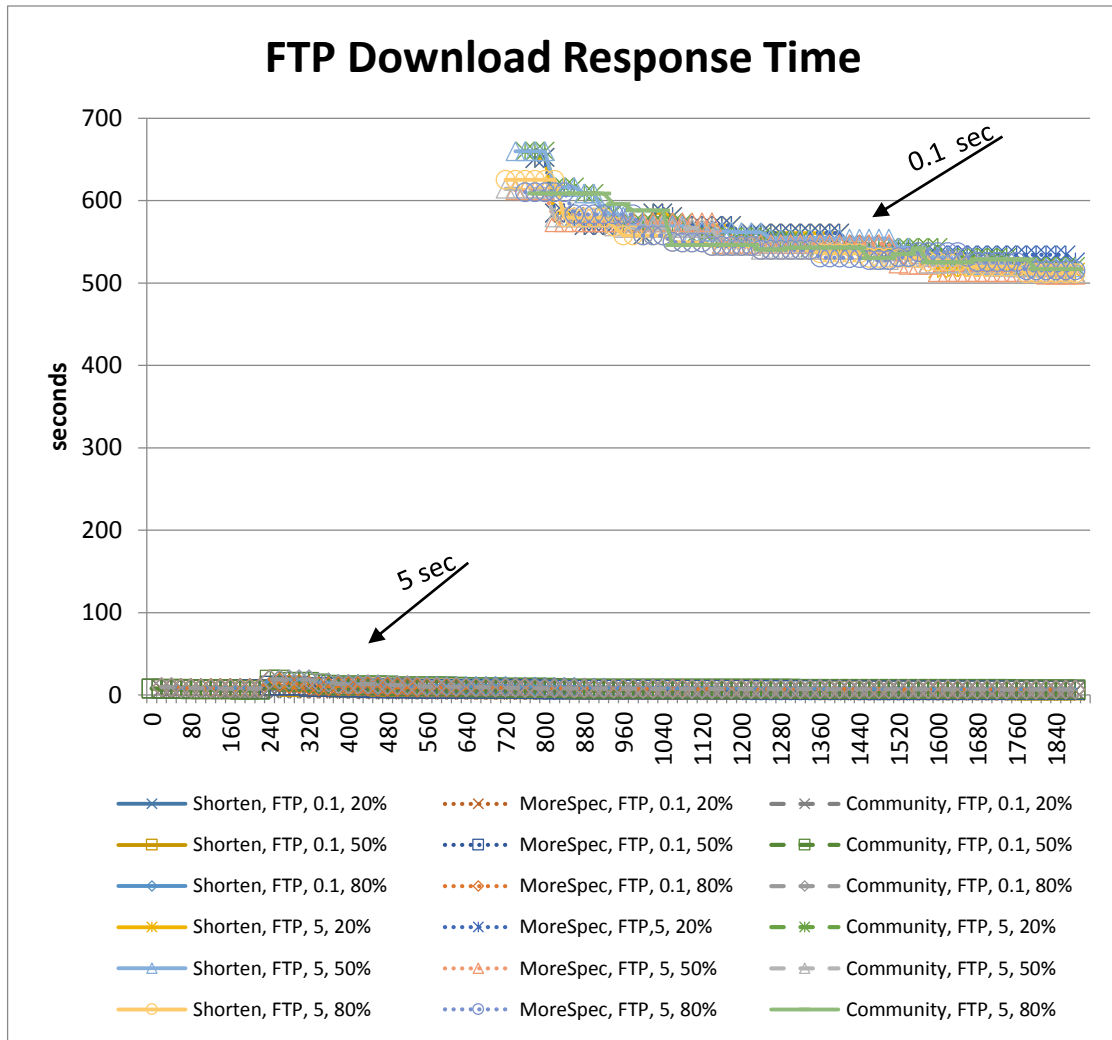


Figure 4.25 FTP Download Response Time.

4.3. Comparison With IPv4

In this section, we compare our results with the IPv4 results obtained by Alrefai[1]. The comparison is with respect to the percentage of dropped packets, the convergence time, and the throughput.

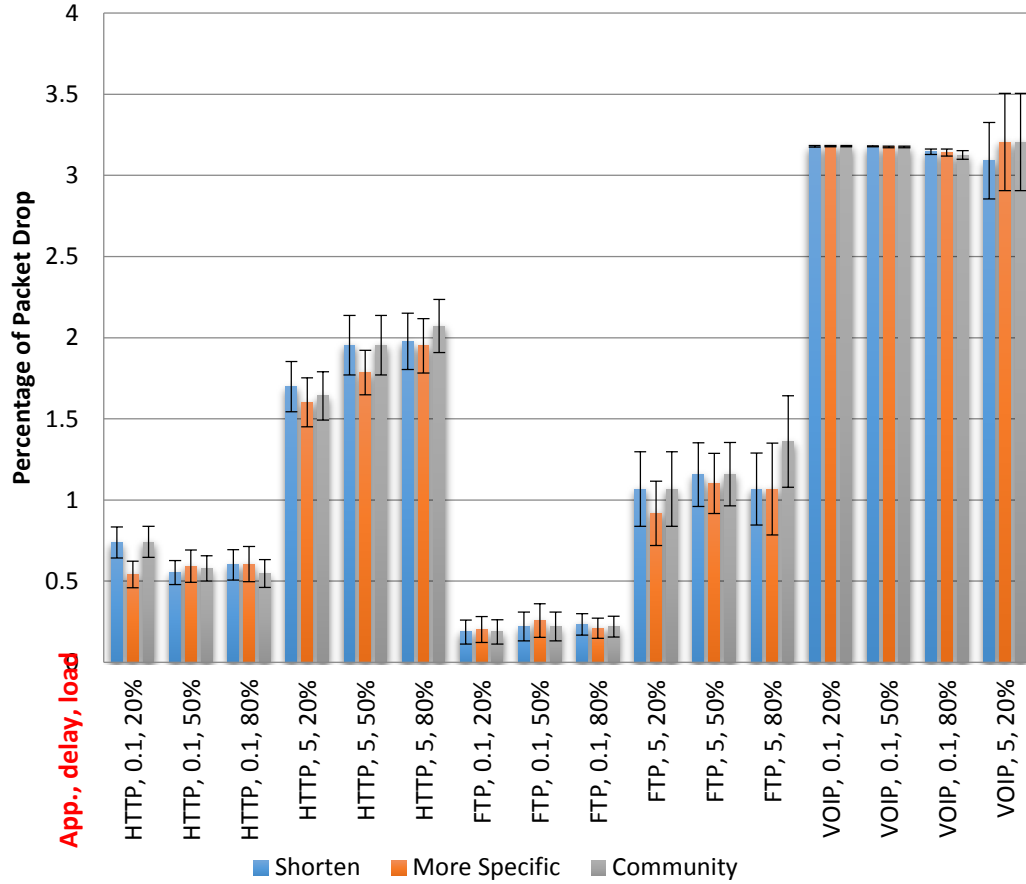


Figure 4.26 Packet drop percentages [1].

Figure 4.26 shows the percentage of dropped packets as obtained by Alrefai [1].

Comparing Figure 4.3 with Figure 4.26 we notice that there are less packet drops in our study. Our study shows a 14% improvement in the percentage of the dropped packets percentage over the IPv4 percentage of the dropping packets. Moreover, our study shows an improvement of %19 in the percentage of the dropped packets over IPv4

results for VoIP application. This might be due to the better performance of IPv6 specially with the real time application, and the improved handling of IPv6 of big files as compared to IPv4 [17].

On the other hand, when comparing the convergence time in our study against that of Alrefai [1] we notice that the results differ dependent on the Internet delay as shown in Figure 4.4 and Figure 4.5 in this study, and Figure 4.27 and Figure 4.28 as obtained by Alrefai [1].

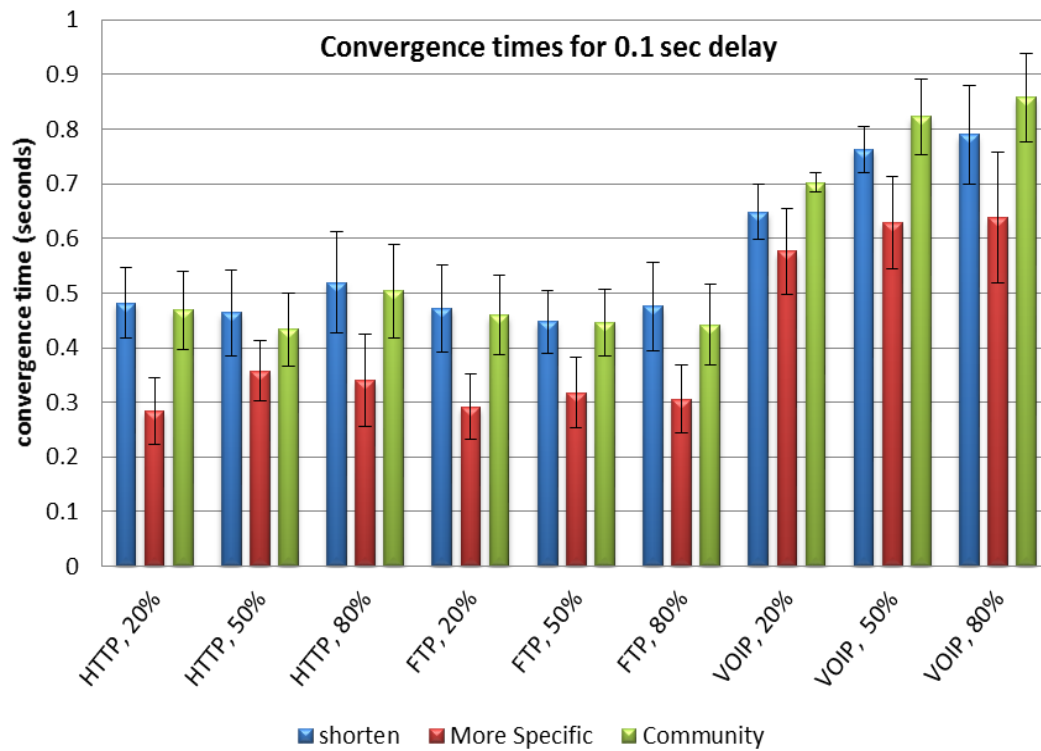
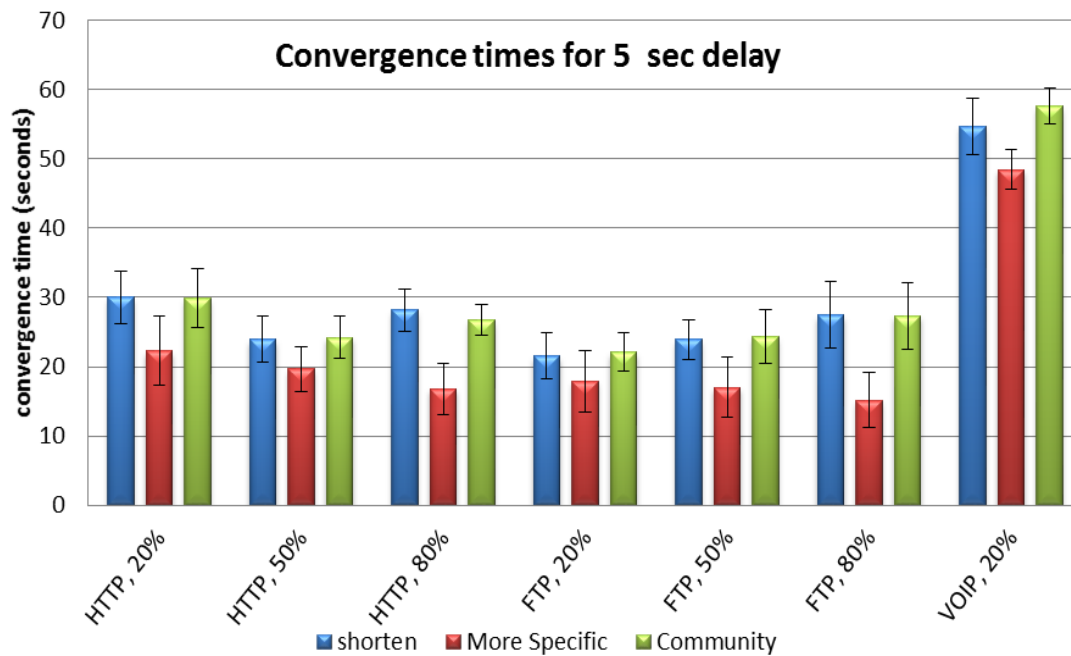


Figure 4.27 BGP convergence time for 0.1 seconds delay of Internet[1].



[Figure 4.28 BGP convergence time for 5 second delay [1].]

It can be noticed that there is an increase in the convergence time in IPv6 when compared against the IPv4 results obtained by Alrefai [1] for the 0.1 second Internet delay. More specifically, our study shows that there is an increase in convergence time for the 0.1 seconds by about 12% for HTTP application, about 7% for the FTP, and 13% for the VoIP application over the results of IPv4. This increase might be due to the large size of the address space of IPv6 which affects the time it takes to update the routers tables. In contrast, when the Internet delay is 5 seconds we notice that the convergence time in our study mostly matches that of Alrefai [1] except for VoIP where our study shows better convergence by about 20%. The reason behind such a behavior is that the 5 seconds Internet delay dominates the extra convergence time associated with IPv6 routers tables updates. Moreover, the IPv6 convergence time for VoIP is smaller than

that for IPv4 because of the better performance of IPv6 with respect to real time applications [19].

The shortening and community solutions perform better in IPv6 than IPv4 for about 12%. This mostly due to the improved structure of router tables under IPv6 which results in less prefix selection time. Subsequently, community and shorten solutions under IPv6 takes less time than IPv4.

The throughput for all applications in our study is slightly higher than those reported by Alrefai [1]. Our study shows increase in the throughput by 3% for HTTP application, 25% for FTP application, and 21% for the VoIP. This is mainly because of the higher packet size of IPv6 as compared to the IPv4 packet size.

4.4. Cases When The Simulation Fails

There are some unexpected behavior for the experiments with the VoIP application. The experiments for the VoIP application with an Internet delay of 5 seconds and for most scenarios of 50% link load and all the scenarios of 80% link load, the traffic switches back and forth between the malicious and the non-malicious routers. This case and others are shown in Figure 4.26.

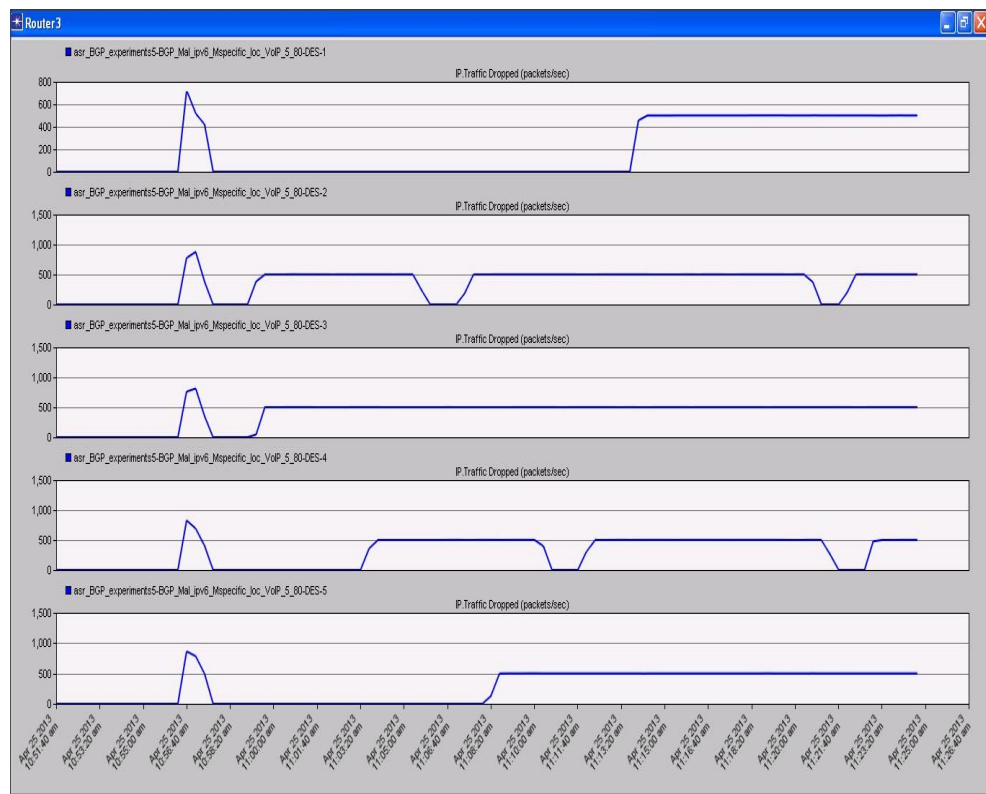


Figure 4. 29 Packets Drop in VoIP, More Specific solution, Exponential with 5 second delay, 80 link load.

This behavior of the traffic happens mainly due to BGP messages hindering. This is because the Internet node is configured with an exponential delay of 5 seconds as mean for the delay. Accordingly, BGP messages suffer from the high delay and the

load links which result in delayed and out of order BGP messages. The problem is aggravated further by the fact that VoIP is a real time application that requires high traffic demand. For more detailed explanation of the problem see section 6.3 in Alrefai [1].

4.5. Summary

In this chapter, we evaluated the performance of using the BGP tuning techniques to solve the Internet access denial problem that is caused by malicious ISP. The solution type, delay of Internet, type of application, and the load in terms of putting more load on specific links are the factors studied in our simulation. The percentage of packet drop, convergence time, throughput, application packet sent and received, page response time, and download response time are the metrics used in the evaluation. Finally, we compared our results with the results obtained by Alrefai [1].

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1. Conclusion

The goal of this thesis work is to implement, evaluate and compare approaches to solve the problem of Internet access denial that is caused by malicious ISPs in IPv6 networks. All the solutions tested are BGP based solutions. The thesis focuses on three BGP-based techniques to solve the problem; AS-Path shortening, more specific prefix, and community. The thesis then describes the design, implementation, and validation of the BGP tuning techniques used. A performance evaluation is provided for the BGP tuning techniques. The performance is conducted in terms of type of solution, Internet delay, application type, and load. BGP convergence time, throughput, packet drop, and response time are the metrics used for comparison. Based on the results obtained, the more specific prefix method has the lowest convergence time while the shortening and community methods have almost the same convergence time. However, the community method has the lowest dropped packets percentage. All methods have almost the same performance for the throughput. Finally, the results of the performance evaluation were compared against the results obtained by Alrefai [1].

5.2. Future Work

The work done in this thesis can be extended further as follows:

- Evaluate the methods used in this work with different simulation tools such as network simulator (NS3) [16].

- Apply the methods on a prototype system and compare the results with our simulation results.
- Implement virtual peering methods as a solution to the Internet access denial problem and evaluate the performance of these methods for IPv6 network.
- Devise and test solutions using IPv6 specific features such as flow label and multi-hop.

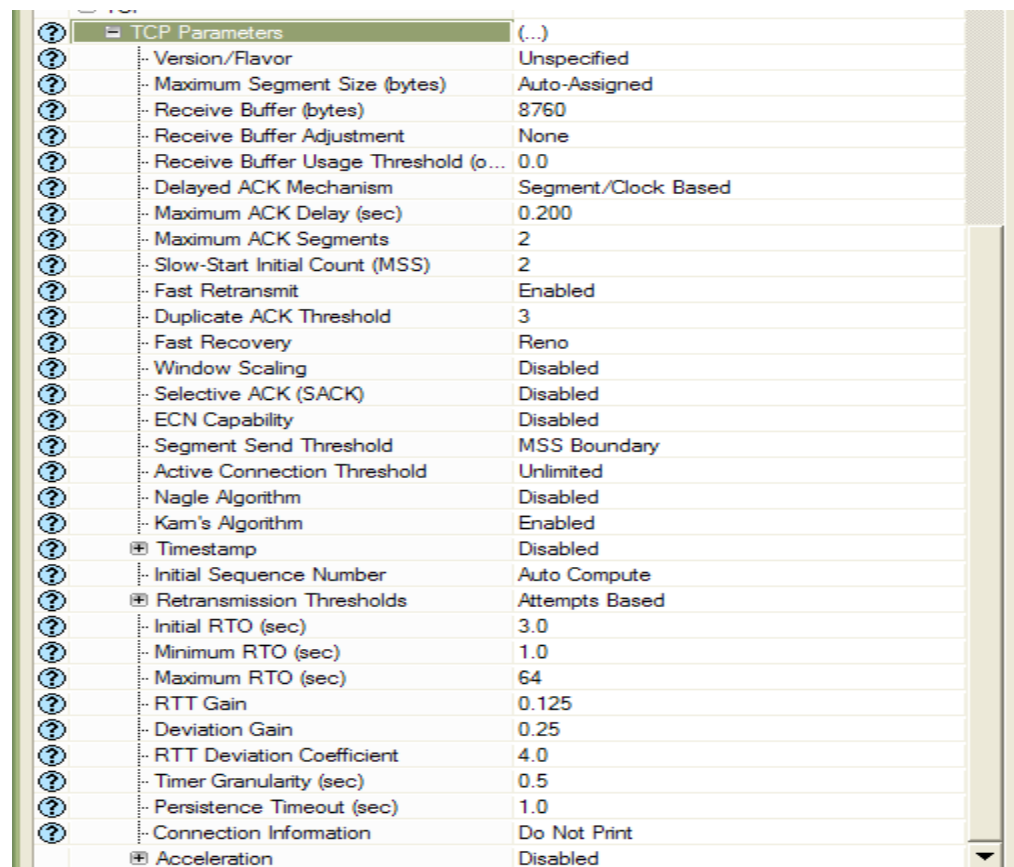
Appendix A

APPLICATION CONFIGURATION

In this appendix we present the configuration of the protocols used in the experiments. The configurations of TCP, HTTP, FTP and VoIP are shown. The configuration shows the default OPNET parameters which we have used.

A.1 TCP configuration

Figure A.1 shows the TCP configuration used.



?	TCP Parameters	(...)
?	Version/Flavor	Unspecified
?	Maximum Segment Size (bytes)	Auto-Assigned
?	Receive Buffer (bytes)	8760
?	Receive Buffer Adjustment	None
?	Receive Buffer Usage Threshold (o...	0.0
?	Delayed ACK Mechanism	Segment/Clock Based
?	Maximum ACK Delay (sec)	0.200
?	Maximum ACK Segments	2
?	Slow-Start Initial Count (MSS)	2
?	Fast Retransmit	Enabled
?	Duplicate ACK Threshold	3
?	Fast Recovery	Reno
?	Window Scaling	Disabled
?	Selective ACK (SACK)	Disabled
?	ECN Capability	Disabled
?	Segment Send Threshold	MSS Boundary
?	Active Connection Threshold	Unlimited
?	Nagle Algorithm	Disabled
?	Kam's Algorithm	Enabled
?	Timestamp	Disabled
?	Initial Sequence Number	Auto Compute
?	Retransmission Thresholds	Attempts Based
?	Initial RTO (sec)	3.0
?	Minimum RTO (sec)	1.0
?	Maximum RTO (sec)	64
?	RTT Gain	0.125
?	Deviation Gain	0.25
?	RTT Deviation Coefficient	4.0
?	Timer Granularity (sec)	0.5
?	Persistence Timeout (sec)	1.0
?	Connection Information	Do Not Print
?	Acceleration	Disabled

Figure A. 1 TCP Configuration.

A.2 HTTP Configuration

Figure A.2 shows the HTTP configuration. HTTP 1.1 is the default version used by OPNET and in our experiments. The page Interarrival time is exponentially distributed with mean 60 seconds.

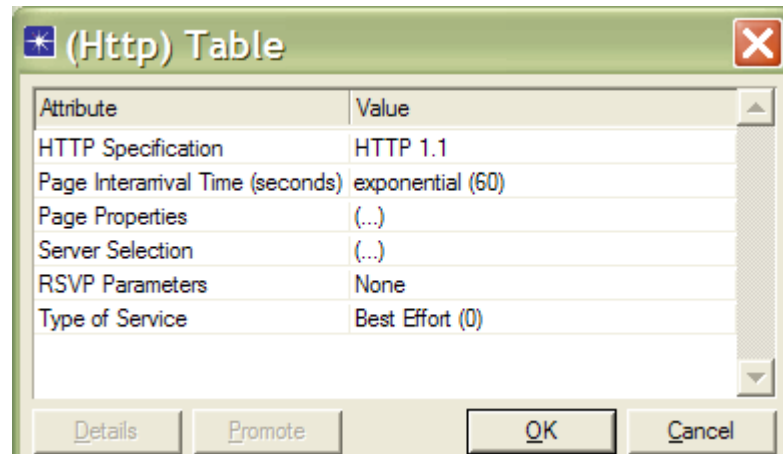


Figure A. 2 HTTP Configuration.

Figure A.3 shows the properties of an HTTP page. Each page has five medium sized images in addition to 1000 bytes page size.

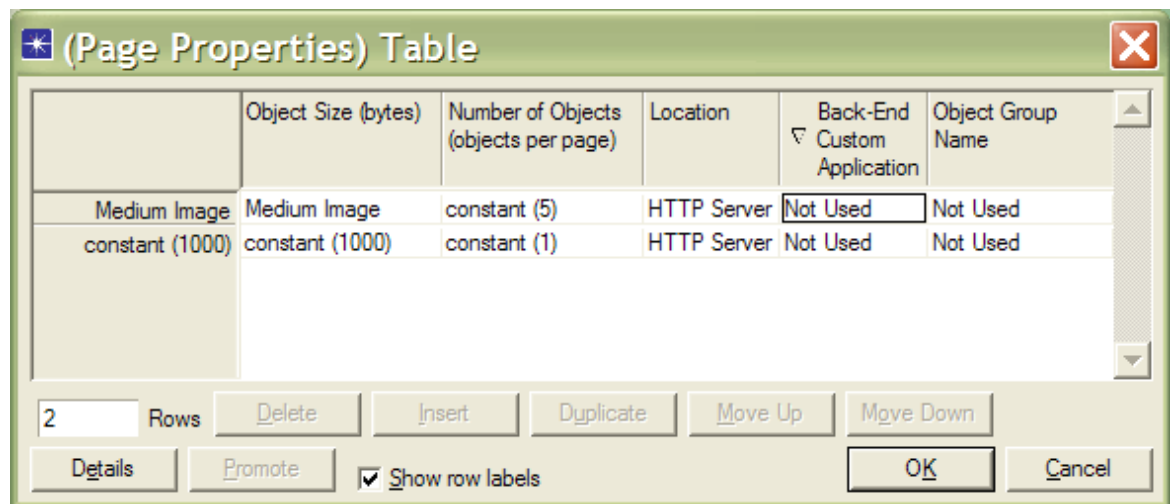


Figure A. 3 HTTP Page Properties.

The image used has a uniform distribution between 500 and 2000 bytes and is shown in Figure A.4.

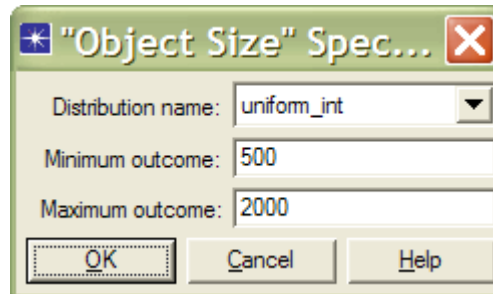


Figure A. 4 Size of Image.

Figure A.5 shows the server selection where the number of pages per server was set to be exponentially distributed with mean 10 pages.

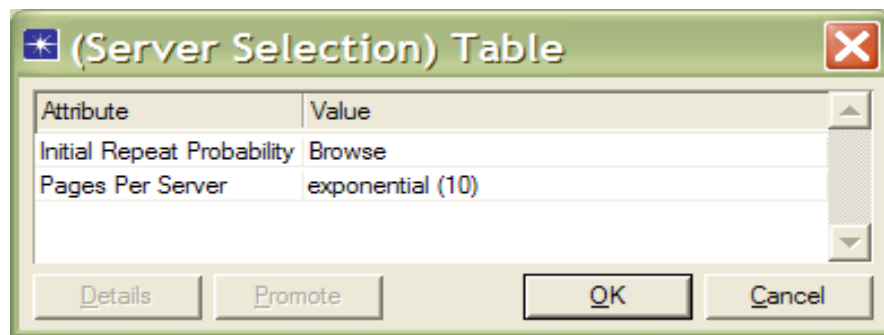


Figure A. 5 HTTP Server Selection.

A.3 FTP Configuration

The FTP configuration table is shown in Figure A.6. The Inter-Request time is exponentially distributed with mean 360 seconds. The Get command is 50% of total commands. The downloaded file size is 50000 bytes.

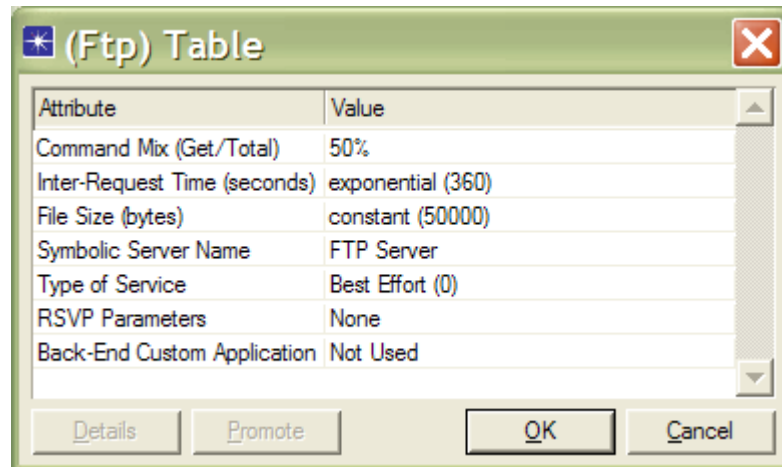


Figure A. 6 FTP Configuration.

A.4 VoIP Configuration

Figure A.7 shows the VoIP configurations table. As shown, there is one voice frame per packet and the GSM FR encoding scheme is used. The silence length and talk spurt length use the default values that are shown in Figure A.8 and A.9, respectively. Both incoming and outgoing are exponentially distributed with mean 0.65 seconds. Also, both incoming and outgoing talk spurt length is exponentially distributed with a mean of 0.352 seconds.

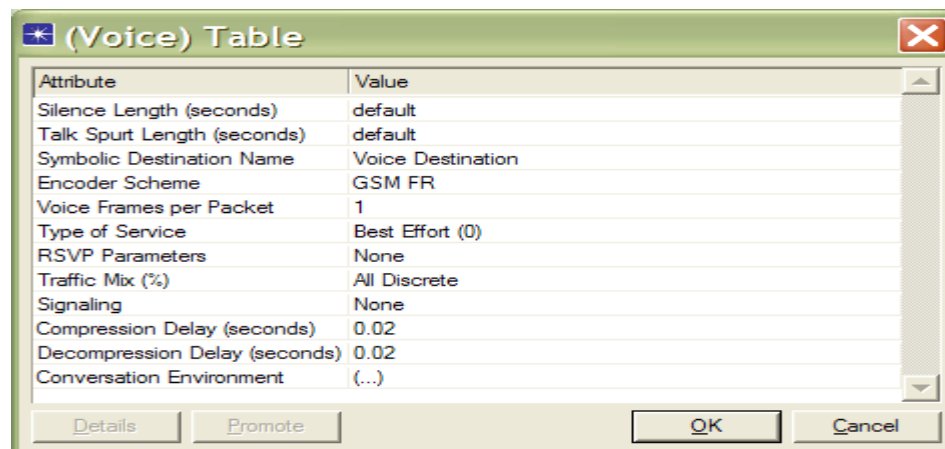


Figure A. 7 VoIP Configuration.

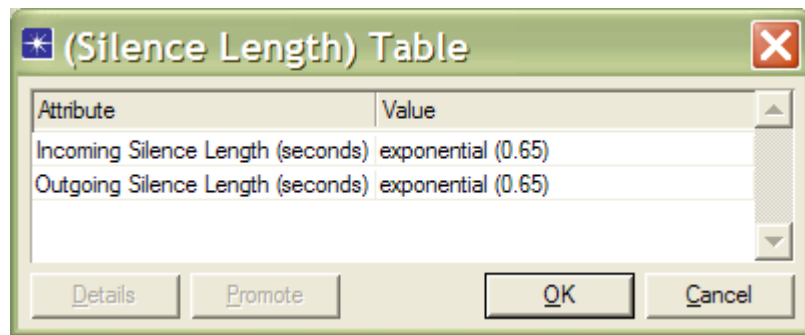


Figure A. 9 Silence Length configuration.

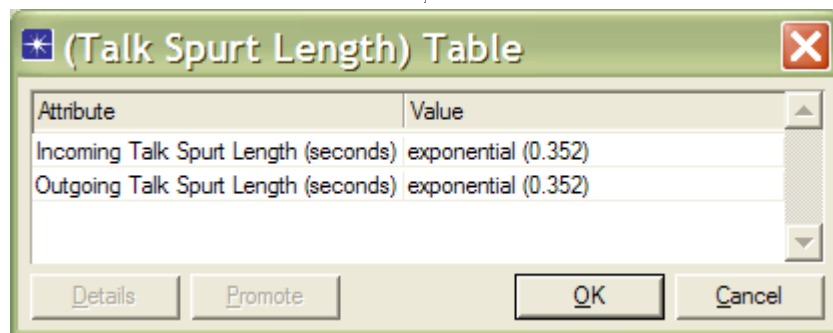


Figure A. 8 Talk Spurt Length

APPENDIX B

BASELINE THROUGHPUT

In this appendix we present the results for the base line simulations. The base line simulations are for HTTP, FTP and VoIP for 0.1 second and 5 seconds Internet delay without any malicious activity. The traffic of the our model without any malicious activity passes through Router3. Thus, we show only the throughput between Router2 and Router in both directions.

Figure B.1 shows the throughput between Router2 and Router3 for HTTP traffic.

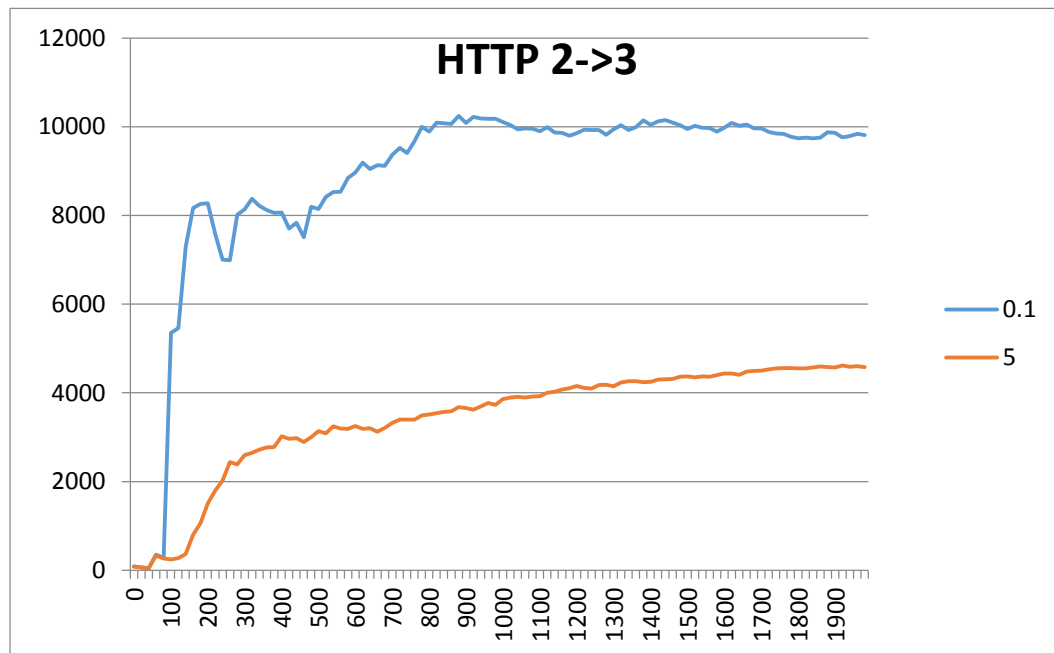


Figure B. 1 Baseline HTTP Throughput from Router2 to Router3.

Figure B.2 shows the throughput from Router3 to Router2 for HTTP traffic.

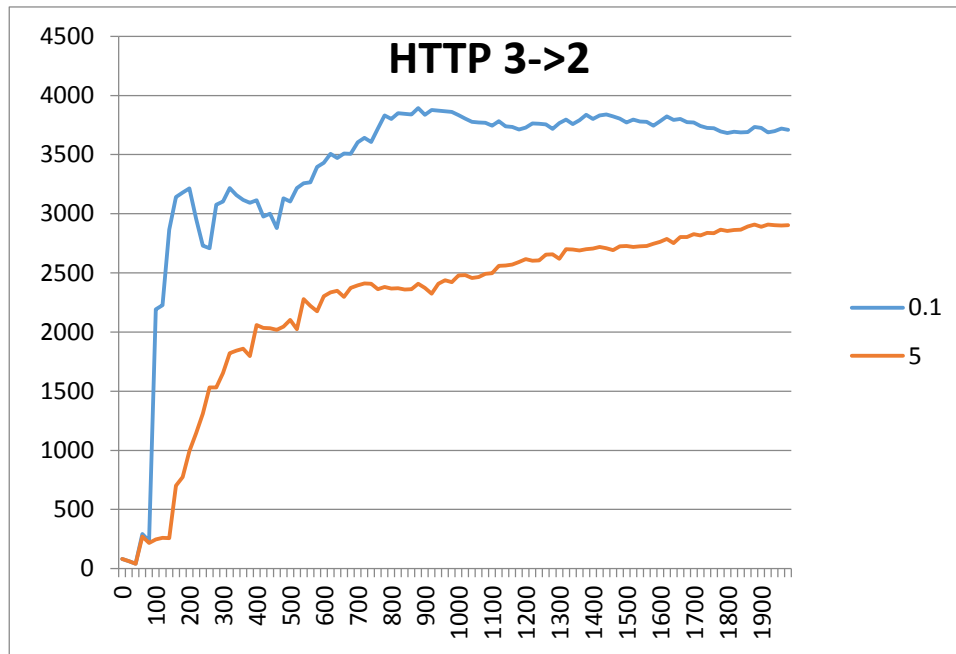


Figure B. 2 Baseline HTTP Throughput from Router3 to Route2

Figure B.3 shows the baseline throughput for FTP from Router2 to Router3.

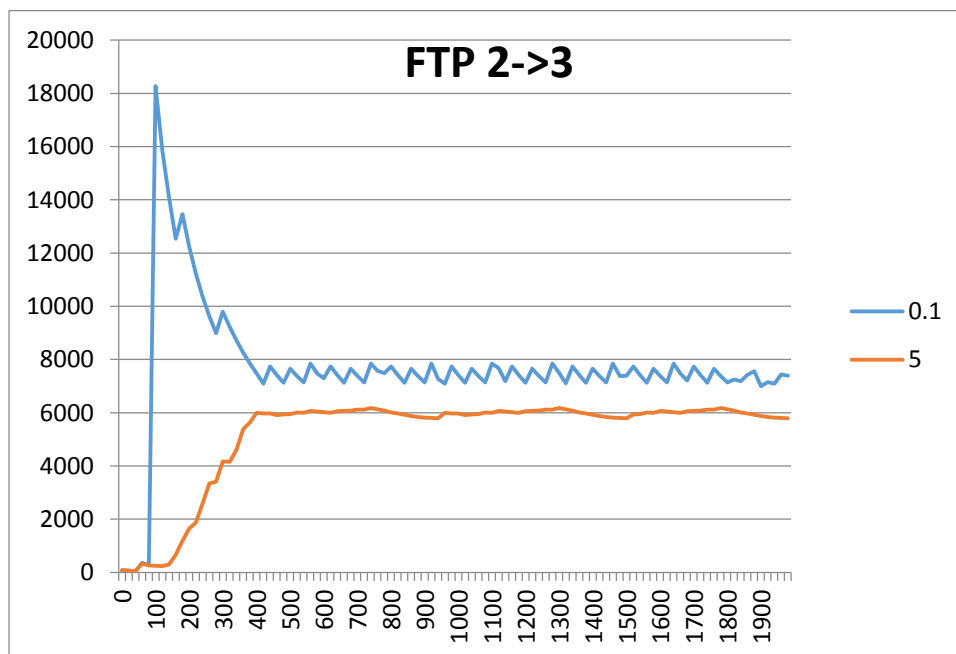


Figure B. 3 Baseline Throughput from Router2 to Router3 for FTP traffic.

Figure B.4 shows the throughput from Router3 to Router2 for FTP traffic.

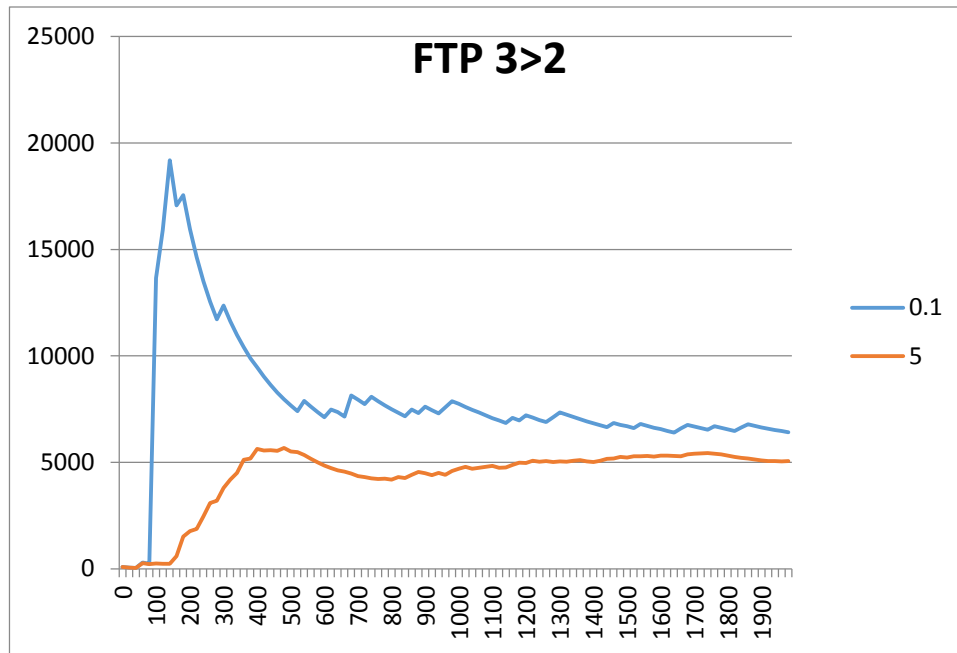


Figure B. 4 Baseline Throughput From Router3 to Router2 for FTP traffic.

Figure B.5 shows the throughput from Router2 to Router3 fro VoIP traffic.

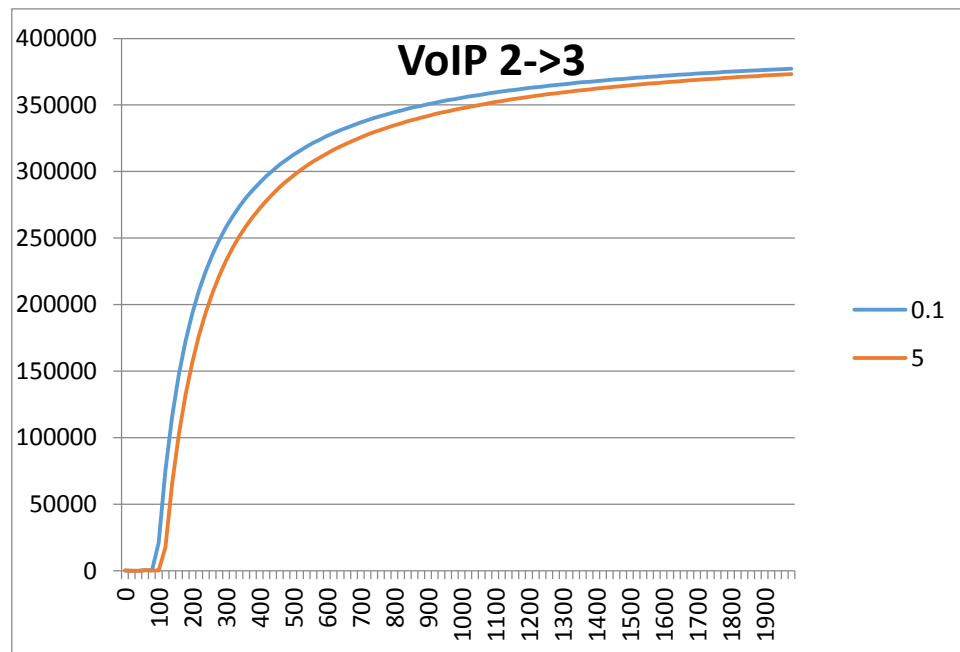


Figure B. 5 Baseline Throughput from Router2 to Router3 for VoIP traffic.

Figure B.6 shows the throughput from Router3 to Router2 for VoIP traffic.

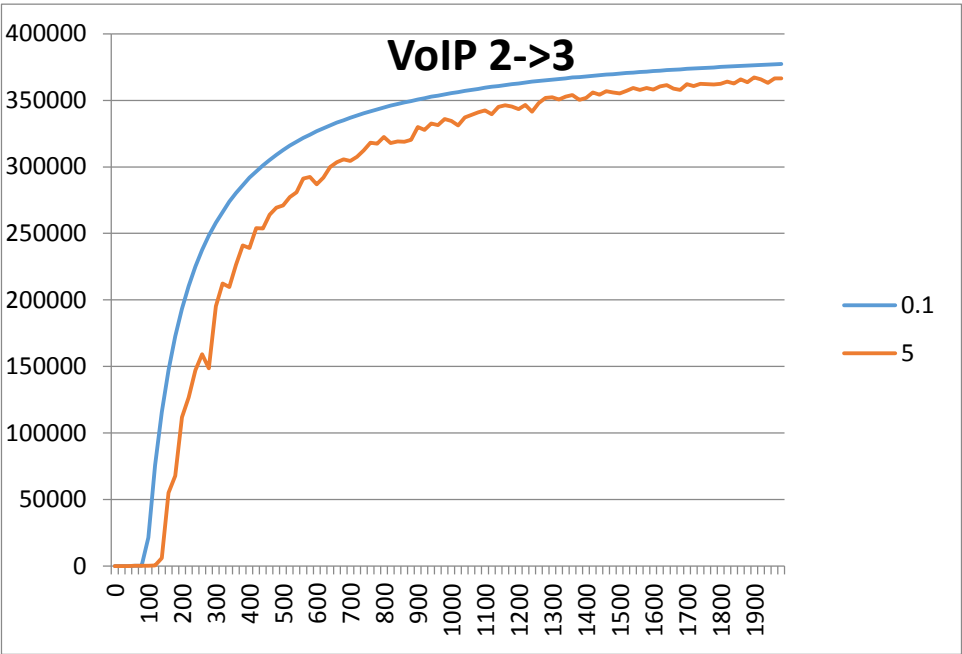


Figure B. 6 Baseline Throughput From Router3 to Router2 for VoIP Traffic.

Appendix C

CODE MODIFICATION

In this appendix we present our code modifications to Alrefai's code to account for IPv6 .

Our Specific modifications are shown underlined in each of the following code segments.

C.1 ReconfigIn State

```
num_entries = op_prg_list_size(bgp_connections_list_ptr);
for(count_i = 0; count_i < num_entries; count_i++ )
{
    bgp_conn_info_ptr = op_prg_list_access (bgp_connections_list_ptr,
count_i);
    if(bgp_conn_info_ptr->neighbor_as_number == a_timed_policy-
>neighbor_as)
        break;
}
op_pro_invoke (bgp_conn_info_ptr->bgp_connection_prohandle,
a_timed_policy->rte_policy_ptr);

/**/
/* almehdhar modifycations to Update State */

if (bgp_conn_info_ptr == OPC_NIL)
{
    /* There is no matching BGP neighbor process that has uses the tcp
    */
    /* Drop the packet. */
    op_pk_destroy (intrpt_info.msg_pkptr);
    intrpt_info.msg_pkptr = OPC_NIL;
}
else
{
    /* Update the statistics on the amount of traffic received.
    */
}
```

```

bgp_traffic_rcvd_stats update ((double) total_size,
received_packet_type);

/* Process found. Invoke the peer and to handle the update message.
*/
op_pro_invoke (bgp_conn_info_ptr->bgp_connection_prohandle,
&intrpt_info);

/* removed from the Loc-RIB.
*/
if (bgp_conn_info_ptr->unreachable_rte_exists == OPC_TRUE)
{
/* Get the list of unreachable routes form the mailbox area.
*/
/*almehdhar codemodivication*/
if (BGPC_ADDR_FAMILY_ATTR IPV4 == tmp_int)
addr_family = BgpC_Ipv4_Address;
else if (BGPC_ADDR_FAMILY_ATTR IPV6 == tmp_int)
addr_family = BgpC_Ipv6_Address;

unreachability_list_ptr = bgp_conn_info_ptr-
>unreachable_rte_list_ptr;
if (ip_node_is_pe (ip_module_data_ptr) &&
(bgp_conn_info_ptr->neighbor_site_vrf_name != OPC_NIL))
bgp_prefix_list_ipv6_to_vpnv4_convert (unreachability_list_ptr,
bgp_conn_info_ptr->neighbor_site_vrf_name);

/* Process the information and re-set the flag.
*/
bgp_unfeasible_routes_process (unreachability_list_ptr,
bgp_conn_info_ptr->peer_id);

/* Make sure that the flag is reset to false so that the next set
of*/
/* unfeasible routes can be properly communicated.
*/
bgp_conn_info_ptr->unreachable_rte_exists = OPC_FALSE;
}

else
{
/* Force the unreachability list pointer to be NULL. This value */
/* will be passed to the procedure that will propagate the new */
/* status of the Local-RIB to all the neighbors. */
unreachability_list_ptr = OPC_NIL;
}
/* Check to see if new routes have been added to the RIB-In
*/
if (bgp_conn_info_ptr->adj_rib_in_ptr->num_new_routes > 0)
{
/* Collect the new routes from the temporary list into the new
*/
/* routes list.
*/
for (count_i = 0; count_i < bgp_conn_info_ptr->adj_rib_in_ptr-
>num_new_routes; count_i++)
{
/* All new routes should be on top of the list. Access the
*/
/* the new routes and add them to the list.
*/
}
}
}

```

```

        /* almehdhar code modification*/
        new_rte_entry_ptr = (BgpT_Rte_Entry*) op_prg_list_remove
(bgp_conn_info_ptr->adj_rib_in_ptr->new_routes_lptr, OPC_LISTPOS_HEAD);

        if (BgpC_Conn_Type_Ebgp == bgp_conn_info_ptr-
>bgp_connection_type)
        {
            new_rte_entry_ptr->admin = admin_distance;
        }
        else
        {
            new_rte_entry_ptr->admin = ibgp_admin_distance;
        }

        /* Check if this new entry is from a VPN site
        */
        if (ip_node_is_pe (ip_module_data_ptr) &&
            (bgp_conn_info_ptr->neighbor_site_vrf_name != OPC_NIL))
        {
            /*almehdhar codemodivication*/

            /* Check if this new entry is from a VPN site. And RD
            and */
            /* values are not set for this route. If it is
            */
            /* then set the route distinguisher value for the entry
            */
            bgp_new_rte_at_vpn_pe_process (new_rte_entry_ptr,
            bgp_conn_info_ptr->neighbor_site_vrf_name);
        }

        op_prg_list_insert (new_rte_list_ptr, new_rte_entry_ptr,
        OPC_LISTPOS_HEAD);

        /* Continue till all the new routes have been inserted.
        */
    }

    /* Call the procedure that will process the new routes.
    */
    bgp_reachability_info_process (bgp_conn_info_ptr);

    /* Reset the number of new routes to 0 and destroy the temporary */
    /* list.
    */
    bgp_conn_info_ptr->adj_rib_in_ptr->num_new_routes = 0;
    op_prg_mem_free (bgp_conn_info_ptr->adj_rib_in_ptr->new_routes_lptr);
}

/* Find out the number of new routes that were entered into the local */
/* routing table. This would not only be the number of routes that
*/
/* that were received as a part of the advertisement, but also could */
/* contain the replacement routes that were selected after certain
*/
/* routes were termed infeasible.
*/

```



```

number_of_new_routes = op_prg_list_size (new_rte_list_ptr);

/* If any of the list is valid, then all the peer processes have to */
/* be notified about the change in the routing table status. */
if ((number_of_new_routes > 0) || unreachable_list_ptr != OPC_NIL)
{
    /* Unless this is a dummy node representing an external AS, */
    /* propagate the new routes to all the other neighbors. */
    /*
    if (OPC_FALSE == is_external_as_node)
    {
        bgp_new_routes_propagate (unreachable_list_ptr,
number_of_new_routes, bgp_conn_info_ptr->peer_id);
    }

    /* Clean up just the new_rte_list_ptr by removing all the route */
    /* entries in it. Be sure not to free up the memory of the */
    /* route entries as these entries are used by the route tables. */
    for (count_i = 0; count_i < number_of_new_routes; count_i++)
    {
        /* remove the routes entries from the new route list. */
        op_prg_list_remove (new_rte_list_ptr, OPC_LISTPOS_HEAD);
    }

    /* Clean up the unreachable routes list. The prefixes in this list */
    /* can be freed up. The will not be reference by any route entry. */
    if (unreachable_list_ptr != OPC_NIL)
    {
        /* Destroy the list of unreachable routes. */
        /*
        bgp_support_rte_list_destroy (unreachable_list_ptr);
        bgp_conn_info_ptr->unreachable_rte_list_ptr = OPC_NIL;
        */
    }
}

```

C.2 ReconfigOut State

```
/* you still need to the update message must include the original*/

num_entries = op_prg_list_size(bgp_connections_list_ptr);
for(count_i = 0; count_i < num_entries; count_i++)
{
    bgp_conn_info_ptr = op_prg_list_access (bgp_connections_list_ptr,
count_i);
    if(bgp_conn_info_ptr->neighbor_as_number == a_timed_policy-
>neighbor_as)
        break;
}
rte_list_ptr = bgp_conn_info_ptr->adj_rib_out_ptr->rte_list_ptr;
num_entries = op_prg_list_size (rte_list_ptr);
new_rte_list_ptr = op_prg_list_create();
if (! a_timed_policy->isMoreSpecific)
{
    bgp_conn_info_ptr->adj_rib_out_ptr->new_routes_lptr =
op_prg_list_create ();
    bgp_conn_info_ptr->adj_rib_out_ptr->num_new_routes = 0;
    unreachable_list_ptr = op_prg_list_create();
    for (count_i = 0; count_i < num_entries; count_i++)
    {
        rte_entry_ptr = (BgpT_Rte_Entry*) op_prg_list_access
(rte_list_ptr, count_i);
        new_rte_entry_ptr = bgp_support_rte_entry_copy(rte_entry_ptr);
        rte_maps = op_prg_list_create();
        op_prg_list_insert(rte_maps, a_timed_policy->rte_policy_ptr,
OPC_LISTPOS_TAIL);

        if(bgp_support_rte_filter_policy_apply(&new_rte_entry_ptr,    rte_maps,
OPC_NIL, OPC_NIL, OPC_FALSE, &policy_edited,
```

```

bgp_conn_info_ptr->bgp_connection_type, bgp_conn_info_ptr-
>local_info_ptr) == OPC_TRUE)
{
    if (policy_edited == OPC_TRUE)
    {
        bgp_support_rte_entry_print(rte_entry_ptr);
        new_mp_prefix_ptr = (BgpT_Mp_Prefix*)
bgp_support_mp_prefix_copy(rte_entry_ptr->dest_prefix_ptr);
        if(new_mp_prefix_ptr == OPC_NIL)
            bgp_support_mp_prefix_print (prefix_str, new_mp_prefix_ptr);
        op_prg_list_insert (bgp_conn_info_ptr->unreachable_rte_list_ptr
,new_mp_prefix_ptr, OPC_LISTPOS_HEAD);
    if (BGPC_ADDR_FAMILY_ATTR_IPV4 == tmp_int)
        addr_family = BgpC_Ipv4_Address;
        else if (BGPC_ADDR_FAMILY_ATTR_IPV6 == tmp_int)
            addr_family = BgpC_Ipv6_Address;

        bgp_conn_info_ptr->unreachable_rte_exists = OPC_TRUE;
        bgp_support_ith_rte_entry_replace (bgp_conn_info_ptr-
>adj_rib_out_ptr, count_i, new_rte_entry_ptr);
        op_prg_list_insert (new_rte_list_ptr,
new_rte_entry_ptr, OPC_LISTPOS_TAIL);
        bgp_conn_info_ptr->adj_rib_out_ptr->num_new_routes++;
        op_prg_list_insert (bgp_conn_info_ptr->adj_rib_out_ptr-
>new_routes_lptr,new_rte_entry_ptr, OPC_LISTPOS_TAIL);
    }
}
else
{
    /* Restrict this route.
*/
        bgp_support_rte_entry_destroy (new_rte_entry_ptr);
        /* Update the statistics that indicate the number */
        /* routes that were dropped due to route policies. */
op_stat_write (bgp_conn_info_ptr->local_info_ptr-
>bgp_local_stats.num_policy_discards_local_stat_hndl, 1.0);
        op_stat_write (bgp_conn_info_ptr->local_info_ptr-
>bgp_local_stats.num_policy_discards_local_stat_hndl, 0.0);
    }
}

if (ip_node_is_pe (ip_module_data_ptr) &&
(bgp_conn_info_ptr->neighbor_site_vrf_name != OPC_NIL))
    bgp_prefix_list_ipv6_to_vpnv6_convert (bgp_conn_info_ptr-
>unreachable_rte_list_ptr, bgp_conn_info_ptr->neighbor_site_vrf_name);
number_of_new_routes = op_prg_list_size (new_rte_list_ptr);
    bgp_support_rte_table_print (bgp_conn_info_ptr->adj_rib_out_ptr);

```

```

/* Clean up just the new_rte_list_ptr by removing all the route      */
/* entries in it. Be sure not to free up the memory of the          */
/*
/* route entries as these entries are used by the route tables.
*/
/* remove the routes entries from the new route list.
*/

{
    op_pro_invoke (bgp_conn_info_ptr->bgp_connection_prohandle,
OPC_NIL);
    for (count_i = 0; count_i < number_of_new_routes; count_i++)
    {
        ///< remove the routes entries from the new route list.

        op_prg_list_remove (new_rte_list_ptr, OPC_LISTPOS_HEAD);
    }
    if(OPC_NIL != bgp_conn_info_ptr->unreachable_rte_list_ptr)
    {
        bgp_support_rte_list_destroy(bgp_conn_info_ptr->
unreachable_rte_list_ptr);
        bgp_conn_info_ptr->unreachable_rte_list_ptr = OPC_NIL;
        bgp_conn_info_ptr->unreachable_rte_exists = OPC_FALSE;
    }
}
else
{
    /* Here we want to handle more specific prefixes*/

    bgp_conn_info_ptr->adj_rib_out_ptr->new_routes_lptr =
op_prg_list_create();

    bgp_conn_info_ptr->adj_rib_out_ptr->num_new_routes = 0;
    for(count_i = 0; count_i < number_of_new_routes; count_i++)
    {
        op_prg_list_insert(bgp_conn_info_ptr->adj_rib_out_ptr->
new_routes_lptr,
        bgp_support_rte_entry_copy((BgpT_Rte_Entry*)
op_prg_list_access(new_rte_list_ptr, count_i)), OPC_LISTPOS_TAIL);
        ///< bgp support rte entry print((BgpT_Rte_Entry*)
op_prg_list_access(bgp_conn_info_ptr->adj_rib_out_ptr->new_routes_lptr,
OPC_LISTPOS_TAIL));
        ///< bgp_conn_info_ptr->adj_rib_out_ptr->num_new_routes++;
    }

    bgp_conn_info_ptr->adj_rib_out_ptr->num_new_routes =
op_prg_list_size(new_rte_list_ptr);
    if(bgp_conn_info_ptr->adj_rib_out_ptr->num_new_routes > 0 ||
bgp_conn_info_ptr->unreachable_rte_exists == OPC_TRUE)

```

```

        /* read the prefix and the number of bits to divide*/
/* search for it in rib out */
/* if found store route attribute */
/* divide it into list of prefixes */
/* create the route with the same path attribute of rib out */
/* add the routes to rib out */
/* send it to the specific neighbor by invoking the process!! */

mp_prefix_ptr = a_timed_policy->prefix_ptr;
rte_entry_ptr = bgp_support_rte_entry_find (bgp_conn_info_ptr-
>adj_rib_out_ptr, mp_prefix_ptr, &location);

if (rte_entry_ptr != OPC_NIL)
{
    num_prefixes = op_prg_list_size (a_timed_policy-
>mp_prefixes_list);
    if (num_prefixes > 0)
    {
        bgp_conn_info_ptr->adj_rib_out_ptr->new_routes_lptr =
op_prg_list_create();

        bgp_conn_info_ptr->adj_rib_out_ptr->num_new_routes = 0;
        for (count_i = 0; count_i < num_prefixes; count_i++)
        {
            new_mp_prefix_ptr = op_prg_list_access
(a_timed_policy->mp_prefixes_list, count_i);
            new_rte_entry_ptr =
bgp_support_rte_entry_copy(rte_entry_ptr);
            new_rte_entry_ptr->dest_prefix_ptr =
new_mp_prefix_ptr;

            op_prg_list_insert(bgp_conn_info_ptr-
>adj_rib_out_ptr->new_routes_lptr, new_rte_entry_ptr, OPC_LISTPOS_TAIL);
            bgp_conn_info_ptr->adj_rib_out_ptr-
>num_new_routes++;
            bgp_support_rte_entry_insert (bgp_conn_info_ptr-
>adj_rib_out_ptr, new_rte_entry_ptr);
        }
        if(bgp_conn_info_ptr->adj_rib_out_ptr-
>num_new_routes > 0)
        {
            op_pro_invoke (bgp_conn_info_ptr-
>bgp_connection_prohandle, OPC_NIL);
            for (count_i = 0; count_i < number_of_new_routes;
count_i++)
            {
                // remove the routes entries from the new
route list.
                op_prg_list_remove (new_rte_list_ptr,
OPC_LISTPOS_HEAD);
            }
        }
    }
}

}

```

C.3 Changes in BGP Module

Modification in bgp_conn Process

```
static void bgp_conn_apply_map_rib_in (IpT_Rte_Policy* rte_policy_ptr)
{
    List* rte_maps;
    Boolean edit_status;
    List* rte_list_ptr;
    BgpT_Rte_Entry*   rte_entry_ptr;
    BgpT_Rte_Entry*   new_rte_entry_ptr;
    int               num_entries;
    int               count_i;
    /* what needed to be done
    1. loop the rib in      2. copy each entry
    3. apply policy         4. if accepted process the new route
    */
    FIN(bgp_conn_apply_map_rib_in (rte_policy_ptr));
    rte_list_ptr = bgp_my_adj_rib_in_ptr->rte_list_ptr;
    num_entries = op_prg_list_size (rte_list_ptr);
    if (BGPC_ADDR_FAMILY_ATTR_IPV4 == tmp_int)
        addr_family = BgpC_Ipv4_Address;
    else if (BGPC_ADDR_FAMILY_ATTR_IPV6 == tmp_int)
        addr_family = BgpC_Ipv6_Address;

    /** Convert a list of IP prefixes into MP-prefixes.**/
    for (ith_prefix = 0; ith_prefix < num_prefixes; ith_prefix++)
    {
        ip_prefix_ptr = (BgpT_Ip_Prefix *) op_prg_list_remove
        (prefix_lptr, ith_prefix);

        op_prg_list_insert (prefix_lptr,
        bgp_support_mp_prefix_from_ip_prefix (ip_prefix_ptr),
        ith_prefix);
    }
    bgp_my_adj_rib_in_ptr->new_routes_lptr = op_prg_list_create
    ();
    for (count_i = 0; count_i < num_entries; count_i++)
    {
        rte_entry_ptr = (BgpT_Rte_Entry*) op_prg_list_access
        (rte_list_ptr, count_i);
        new_rte_entry_ptr = bgp_support_rte_entry_copy(rte_entry_ptr);
        /* because the method of applying policy only accept list of policies
        we need to create a list and insert rte_policy_ptr to it.*/
        rte_maps = op_prg_list_create();
        op_prg_list_insert(rte_maps, rte_policy_ptr, OPC_LISTPOS_TAIL);
    }
}
```

```

        if(bgp_support_rte_filter_policy_apply(&new_rte_entry_ptr,
rte_maps, OPC_NIL, OPC_NIL, OPC_TRUE, &edit_status, BgpC_Conn_Type_None,
conn_info_ptr->local_info_ptr) == OPC_TRUE)
        {
            if(edit_status)
            {
                bgp_conn_route_entry_process(new_rte_entry_ptr);
            }
        }
    else
    {
        //bgp_conn_previously_advertised_route_check
(new_rte_entry_ptr->dest_prefix_ptr);
        //bgp_support_rte_entry_destroy (new_rte_entry_ptr);
    }
}
if (bgp_my_adj_rib_in_ptr->num_new_routes == 0)
    op_prg_mem_free (bgp_my_adj_rib_in_ptr->new_routes_lptr);
FOUT;
}

```

C.4 Shortening

```
static void bgp_support_as_path_prepend (BgpT_Path_Attrs*
path_attrs_ptr, const IpT_Rte_Map_AsPath_List* as_list_ptr) {

    int*          new_segment_value_array;
    int           as_seg_index;
    int           as_list_index;
    int           seg_length;
    BgpT_Path_Segment* path_segment_ptr;

    /** Prepend the ASes specified in the list to the AS Path.
    **/

    FIN (bgp_support_as_path_prepend (path_attrs_ptr,
as_list_ptr));
/*almehdhar codemodivication*/

    printf("Heh I am inside prepending\n");

    if (0 == as_list_ptr->num_as_numbers)
    {
        printf("calling the remove first fcn\n");
        bgp_support_as_path_remove_first(path_attrs_ptr);
        FOUT;
    }

    /* Add the ASes to the segment of type AS Sequence. This
    */
    /* would be the first element in the list.
    */
    path_segment_ptr = (BgpT_Path_Segment *)
        op_prg_list_access (path_attrs_ptr->as_path_list_ptr,
        OPC_LISTPOS_HEAD);

    /* Find out the number of AS Numbers in this segment.
    */
    seg_length = path_segment_ptr->segment_length;

    /* Create a new array to hold the combined as path
    */
    new_segment_value_array = (int*) prg_cmo_alloc
(bgp_as_path_list_cmh,
        (seg_length + as_list_ptr->num_as_numbers) * sizeof
(int));

    /* Copy the new AS Numbers from the as list to the
    beginning*/
    /* of the new array.
    */
    for (as_list_index = 0; as_list_index < as_list_ptr-
>num_as_numbers; as_list_index++)
    {
        new_segment_value_array [as_list_index] = as_list_ptr-
>as_number_array [as_list_index];
    }
}
```



```

        /* Free up the Old segment value array.
*/
        if (seg_length > 0)
            op_prg_mem_free (path_segment_ptr->segment_value_array);

        /* Set the new value and increment the segment length.      */
        path_segment_ptr->segment_value_array = new_segment_value_array;
        path_segment_ptr->segment_length +=  as_list_ptr->num_as_numbers;

        /* Increment the as path length.
*/
        path_attrs_ptr->as_path_length +=  as_list_ptr->num_as_numbers;

        /* Done with adding. Exit the function
*/
        FOUT;
    }

/* Copy the elements of the original array into new array. */
    for (ith_elem = 1; ith_elem < seg_length; ith_elem++)
    {
        printf("ith_elem: %d\n", ith_elem);
        new_segment_value_array [ith_elem - 1] = ith_path_segment_ptr-
>segment_value_array [ith_elem];
        printf("new_segment_value_array: %d \n",
new_segment_value_array [ith_elem - 1]);
        printf("ith_elem: %d\n", ith_elem);
    }
    /* Free up the
    if (seg_length > 0)
        op_prg_mem_free(ith_path_segment_ptr-
>segment_value_array);
    /* set the new value and increment the segment length.
*/
        ith_path_segment_ptr->segment_value_array =
new_segment_value_array;
        (ith_path_segment_ptr->segment_length)--;
        seg_length = ith_path_segment_ptr->segment_length;
        for (ith_elem = 0; ith_elem < seg_length; ith_elem++)
        {
            printf("element#    %d    is    %d\n",    ith_elem,
ith_path_segment_ptr->segment_value_array[ith_elem]);
        }
        --orig_path_attrs_ptr->as_path_length;

        /* Done with adding. Exit the function
*/
        FOUT;
    }

```

```

void
bgp_support_as_path_remove_first (BgpT_Path_Attrs* orig_path_attrs_ptr)
{
    int*                new_segment_value_array;
    int                 ith_elem;
    int                 seg_length;
    BgpT_Path_Segment* ith_path_segment_ptr;

    /** This function the last AS added to the first place of the list
    **/

    FIN (bgp_support_as_path_remove_first (orig_path_attrs_ptr));
    printf("we are inside remove_first :)\n");
    /* Add the new as to the last segment.
    */
    ith_path_segment_ptr = (BgpT_Path_Segment *)
        op_prg_list_access (orig_path_attrs_ptr->as_path_list_ptr,
        OPC_LISTPOS_TAIL);

    /* Find the length of the segment value.
    */
    seg_length = ith_path_segment_ptr->segment_length;
    printf("size: %d\n", seg_length);
    if(seg_length <= 1)
    {
        printf("I am changing the the segment insider");
        ith_path_segment_ptr = bgp_support_path_seg_mem_alloc ();
        ith_path_segment_ptr->segment_type =
        BgpC_Path_Seg_Type_As_Sequence;
        ith_path_segment_ptr->segment_length = 0;
        ith_path_segment_ptr->segment_value_array = OPC_NIL;
        op_prg_list_remove (orig_path_attrs_ptr->as_path_list_ptr,
        OPC_LISTPOS_TAIL);
        op_prg_list_insert (orig_path_attrs_ptr->as_path_list_ptr,
        ith_path_segment_ptr, OPC_LISTPOS_TAIL);
    }
    else
    {
        /* The memeber segment value is a array of AS numbers. Copy */
        /* that into a new array. */
        new_segment_value_array = (int*) prg_cmo_alloc
        (bgp_as_path_list_cmh, (seg_length-1)*sizeof (int));
        for (ith_elem = 0; ith_elem < seg_length; ith_elem++)
        {
            printf("element# %d is %d\n", ith_elem,
            ith_path_segment_ptr->segment_value_array[ith_elem]);
        }
    }
}

```

C.5 More Specific Prefixes

```
/* Alrefai Code Snippet Start */
static void
bgp_neighbor_more_specific_prefix_read (Objid ith_neighbor_info_id,
InetT_Addr_Family addr_family, int neighbor_as_number)
{
    Objid          msp_objid;
    Objid          jth_msp_info_id, jth_prefix_info_id;
    Objid          prefix_id;
    Objid          first_prefix_id;
    Objid          prefixes_id;
    int            num_msps;
    int            count_i, count_j;
/*almehdhar codemodivication*/

    char           addr_str[INETC_ADDR_RANGE_STR_LEN];
    InetT_Address  ntwk_addr, masked_ntwk_addr;
    InetT_Subnet_Mask inet_smask;
    BgpT_Ip_Prefix* prefix_ptr;
    int            num_prefixes;
    double         time;
    Timed_Policy*  timed_policy_ptr;
    BgpT_Mp_Prefix* mp_prefix_ptr;
    int            smask_length;

    FIN (bgp_neighbor_more_specific_prefix_read (ith_neighbor_info_id,
addr_family));

    op_ima_obj_attr_get (ith_neighbor_info_id, "More Specific Prefix",
&msp_objid);
    num_msps = op_topo_child_count (msp_objid, OPC_OBJTYPE_GENERIC);
    if (num_msps > 0)
    {
        timed_policy_ptr = (Timed_Policy*) op_prg_mem_alloc
(sizeof(Timed_Policy));
        timed_policy_ptr->isIn = OPC_FALSE;
        timed_policy_ptr->isMoreSpecific = OPC_TRUE;
        timed_policy_ptr->mp_prefixes_list = op_prg_list_create();
    }
    for (count_i = 0; count_i < num_msps; count_i++)
    {
        jth_msp_info_id = op_topo_child (msp_objid,
OPC_OBJTYPE_GENERIC, count_i);

        op_ima_obj_attr_get (jth_msp_info_id, "Prefix", &prefix_id);
```

```

first_prefix_id = op_topo_child (prefix_id, OPC_OBJTYPE_GENERIC, 0);

/*almehdhar codemodivication*/

    op_ima_obj_attr_get (first_prefix_id, "IP Address", addr_str);
    ntwk_addr = inet_address_create (addr_str, InetC_Addr_Family_v6);
    if (!inet_address_valid (ntwk_addr))
{bgp_invaidd_network_address_log_write (count_i, addr_str); continue;}
    op_ima_obj_attr_get (first_prefix_id, "Mask", addr_str);
    /*alhabib*/
    /* IPv6 address family. If mask is auto-assigned, use a value of */
    /* 64. 64 is the highest permitted mask length (smallest
network */
    /* size for global unicast addresses being currently allocated.
*/
    //addr_family=InetC_Addr_Family_v6;
    if (InetC_Addr_Family_v6 == addr_family ||
BgpC_Ipv6_Address==addr_family)
    {
        if (0 == strcmp (addr_str, BGPC_SUBNET_MASK_AUTO_ASSIGN_STR))
        {
            //inet_smask = inet_smask_from_length_create (64);
            inet_smask = inet_smask_create (addr_str);
        }
        else
        {
            smask_length = atoi (addr_str);

            if ((smask_length < 0) || (smask_length >
IPC_V6_ADDR_LEN))
            {
else{continue;}
            /* Mask the network address with the subnet mask */
            masked_ntwk_addr = inet_address_mask (ntwk_addr, inet_smask);
            inet_address_destroy (ntwk_addr);
            /* Create a prefix and a route entry corresponding to it. */
            /* Use the "fast" version of the function so that we don't */
            /* create a copy of the address and then destroy the */
            /* masked_ntwk_addr variable.
            */
/*almehdhar codemodivication*/

prefix_ptr = inet_address_range_mem_alloc ();
*prefix_ptr = inet_address_range_create_fast (masked_ntwk_addr,
inet_smask);
    mp_prefix_ptr = bgp_support_mp_prefix_from_ip_prefix(prefix_ptr);
    timed_policy_ptr->prefix_ptr =
bgp_support_mp_prefix_copy(mp_prefix_ptr);
op_ima_obj_attr_get (jth_msp_info_id, "Prefixes", &prefixes_id);
    num_prefixes = op_topo_child_count (prefixes_id,
OPC_OBJTYPE_GENERIC);
    for (count_j = 0; count_j < num_prefixes; count_j++)
    {
        jth_prefix_info_id = op_topo_child (prefixes_id,
OPC_OBJTYPE_GENERIC, count_j);
op_ima_obj_attr_get (jth_prefix_info_id, "IP Address",addr_str);
        ntwk_addr = inet_address_create (addr_str,
InetC_Addr_Family_v6);
        if (!inet_address_valid (ntwk_addr))
{bgp_invaidd_network_address_log_write (count_j, addr_str); continue;}
        op_ima_obj_attr_get (jth_prefix_info_id, "Mask", addr_str);

```

```

/* IPv6 address family. If mask is auto-assigned, use a value of */
/* 64. 64 is the highest permitted mask length (smallest
network */
/* size for global unicast addresses being currently
allocated. */
/*almehdhar codemodivication*/

    addr_family=InetC Addr Family v6;
    if (InetC Addr Family v6 == addr_family ||
BgpC Ipv6 Address==addr_family)

    {
        if (0 == strcmp (addr_str, BGPC SUBNET MASK AUTO ASSIGN STR))
        {
            //inet smask = inet smask from length create (64);
            inet smask = inet smask create (addr_str);
        }
        else
        {
            smask length = atoi (addr_str);
            if ((smask length < 0) || (smask length > IPC V6 ADDR LEN))
            {
                bgp_invaidd_subnet_mask_log_write (count_i,
addr_str);
                continue;
            }
            inet smask = inet smask create (addr_str);
        }
        else{continue;}

        masked_ntwk_addr = inet address mask (ntwk_addr, inet smask);
        inet address destroy (ntwk_addr);
        prefix_ptr = inet address range mem alloc ();
        *prefix_ptr = inet address range create fast (masked_ntwk_addr, inet smask);
        mp_prefix_ptr =
bgp_support mp_prefix from ip_prefix(prefix_ptr);
        op_prg_list_insert (timed_policy_ptr->mp_prefixes_list, mp_prefix_ptr,
OPC_LISTPOS_TAIL);

    }

    op_ima_obj_attr_get (jth_msp_info_id, "Time", &time);

    timed_policy_ptr->time = time;

    timed_policy_ptr->neighbor_as = neighbor_as_number;

    op_prg_list_insert (scheduled_reconfigurations,
timed_policy_ptr, OPC_LISTPOS_TAIL);

}

FOUT;

1

```

C.6 Modification in IP protocol

Objid	malicious_blackholing_objid;
Objid	first_blackholing_objid;
Objid	prefixes_objid;
Objid	ith_prefix_objid;
int	num_blackholing;
int	num_prefixes;
IpT_Rte_Blackhole_From*	blackhole_from_ptr;
double	time;
List*	prefixes;
char	addr_str[INETC_ADDR_RANGE_STR_LEN];
InetT Address	ntwk_addr;
InetT Subnet Mask	inet_smask;
int	count_i;
InetT Address_Range*	prefix_ptr;

```

/* read malicious blackholing information */
    op_ima_obj_attr_get(module_data.ip_parameters_objid, "Malicious
Blackholing", &malicious_blackholing_objid);
    num_blackholing = op_topo_child_count (malicious_blackholing_objid,
OPC_OBJTYPE_GENERIC);
    if (num_blackholing > 0)
    {
        first_blackholing_objid = op_topo_child
(malicious_blackholing_objid, OPC_OBJTYPE_GENERIC, 0);
        op_ima_obj_attr_get (first_blackholing_objid, "Time", &time);
        op_ima_obj_attr_get (first_blackholing_objid, "Prefixes",
&prefixes_objid);
        num_prefixes = op_topo_child_count (prefixes_objid,
OPC_OBJTYPE_GENERIC);
        prefixes = op_prg_list_create ();

/*almehdhar code modivicaton*/

        for (count_i = 0; count_i < num_prefixes; count_i++)
        {
            ith_prefix_objid = op_topo_child (prefixes_objid,
OPC_OBJTYPE_GENERIC, count_i);
            op_ima_obj_attr_get (ith_prefix_objid, "IP Address",
addr_str);
            ntwk_addr = inet_address_create (addr_str,
InetC_Addr_Family_v6);
            if (!inet_address_valid (ntwk_addr)) {printf("network invalid");
continue;}
            op_ima_obj_attr_get (ith_prefix_objid, "Mask", addr_str);
            inet_smask = inet_smask_create (addr_str);
            prefix_ptr = inet_address_range_mem_alloc ();
            *prefix_ptr = inet_address_range_create (ntwk_addr, inet_smask);
            op_prg_list_insert (prefixes, prefix_ptr, OPC_LISTPOS_TAIL);
            inet_address_destroy(ntwk_addr);

        }

        blackhole_from_ptr = op_prg_mem_alloc (sizeof
(IpT_Rte_Blackhole_From));
        blackhole_from_ptr->time = time;
        blackhole_from_ptr->prefixes = prefixes;
        module_data.blackhole_from_ptr = blackhole_from_ptr;
    }
    else
    {
        module_data.blackhole_from_ptr = OPC_NIL;
    }

```

```

Boolean ip_rte_blackhole_traffic (IpT_Rte_Module_Data * iprmd_ptr,
Packet * pkptr)
{
    InetT_Address_Range *      blackholed_prefix;
    InetT_Address              dest_address;
    InetT_Address              src_address;

    char                      addr_str[INETC_ADDR_RANGE_STR_LEN];
    InetT_Address              ntwk_addr;

    int num_prefixes;
    IpT_Dgram_Fields* pk_fd_ptr;
    double time;
    List * prefixes;
    int count_i;
    FIN (ip_rte_blackhole_traffic (iprmd_ptr, pkptr));
    if (iprmd_ptr->blackhole_from_ptr == OPC_NIL)
    {
        FRET (OPC_FALSE);
    }
}

```



```

if (op_sim_time() < time)
{
    FRET (OPC_FALSE);
}

prefixes = iprmd_ptr->blackhole_from_ptr->prefixes;
op_pk_nfd_access (pkptr, "fields", &pk_fd_ptr);
dest_address = pk_fd_ptr->addr_str;

dest_address = inet address create (addr_str, InetC Addr Family v6);

src_address = pk_fd_ptr-> addr_str;
src_address = inet address create (addr_str, InetC Addr Family v6);

num_prefixes = op_prg_list_size(prefixes);
printf("number of packets: %d \n", iprmd_ptr->blackhole_from_ptr->number_of_packets);
for(count_i = 0; count_i < num_prefixes; count_i++)
{
    blackholed_prefix = (InetT_Address_Range *)op_prg_list_access
(prefixes, count_i);
    if (inet_address_range_check (dest_address, blackholed_prefix)
== PRGC_TRUE)
    {
        ip_rte_dgram_discard (iprmd_ptr, pkptr, op_pk_ici_get
(pkptr), "Discarded because destination address is blackholed:");
        iprmd_ptr->blackhole_from_ptr->number_of_blackholing++;

        printf("number of blackholing: %d \n", iprmd_ptr->
>blackhole_from_ptr->number_of_blackholing);
        FRET (OPC_TRUE);
    }
    else if (inet_address_range_check (src_address,
blackholed_prefix) == PRGC_TRUE)
    {
        ip_rte_dgram_discard (iprmd_ptr, pkptr, op_pk_ici_get
(pkptr), "Discarded because source address is blackholed:");
        iprmd_ptr->blackhole_from_ptr->number_of_blackholing++;
        printf("number of blackholing: %d \n", iprmd_ptr->
>blackhole_from_ptr->number_of_blackholing);
        FRET (OPC_TRUE);
    }
}

```

Appendix D

FTP THROUGHPUT WITH INTER-REQUEST Time of 60 Seconds

In this appendix we present the results for the FTP throughput with an inter-request time of 60 seconds. The simulations are for 0.1 second and 5 seconds Internet delay.

Figure D.1 shows the FTP throughput between Router 2 and Router 3 when the inter-request time is 60 seconds.

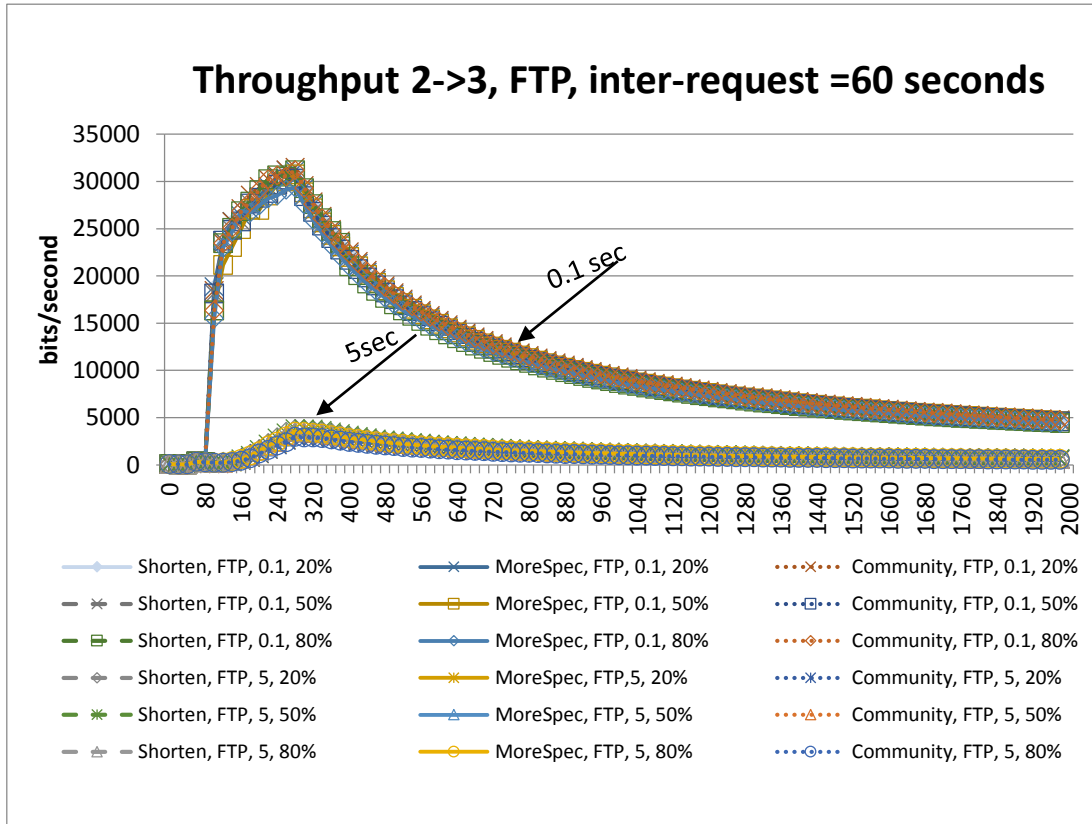


Figure D.1 Throughput for FTP application , inter-request is 60 seconds.

Figure D.2 shows the FTP throughput between Router 3 and Router 2 when the inter-request time is 60 seconds.

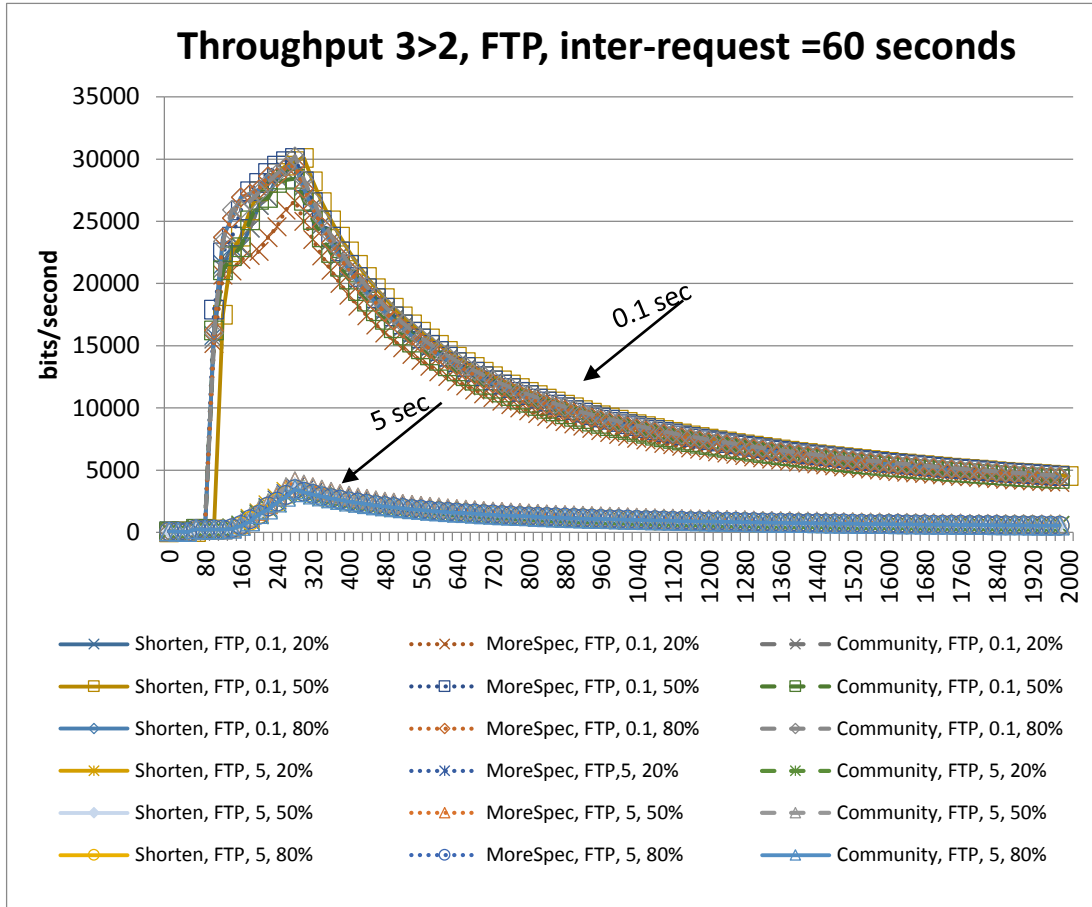


Figure D.2 Throughput for FTP application , inter-request is 60 seconds

Figure D.3 shows the FTP throughput between Router 2 and Router 4 when the inter-request time is 60 seconds.

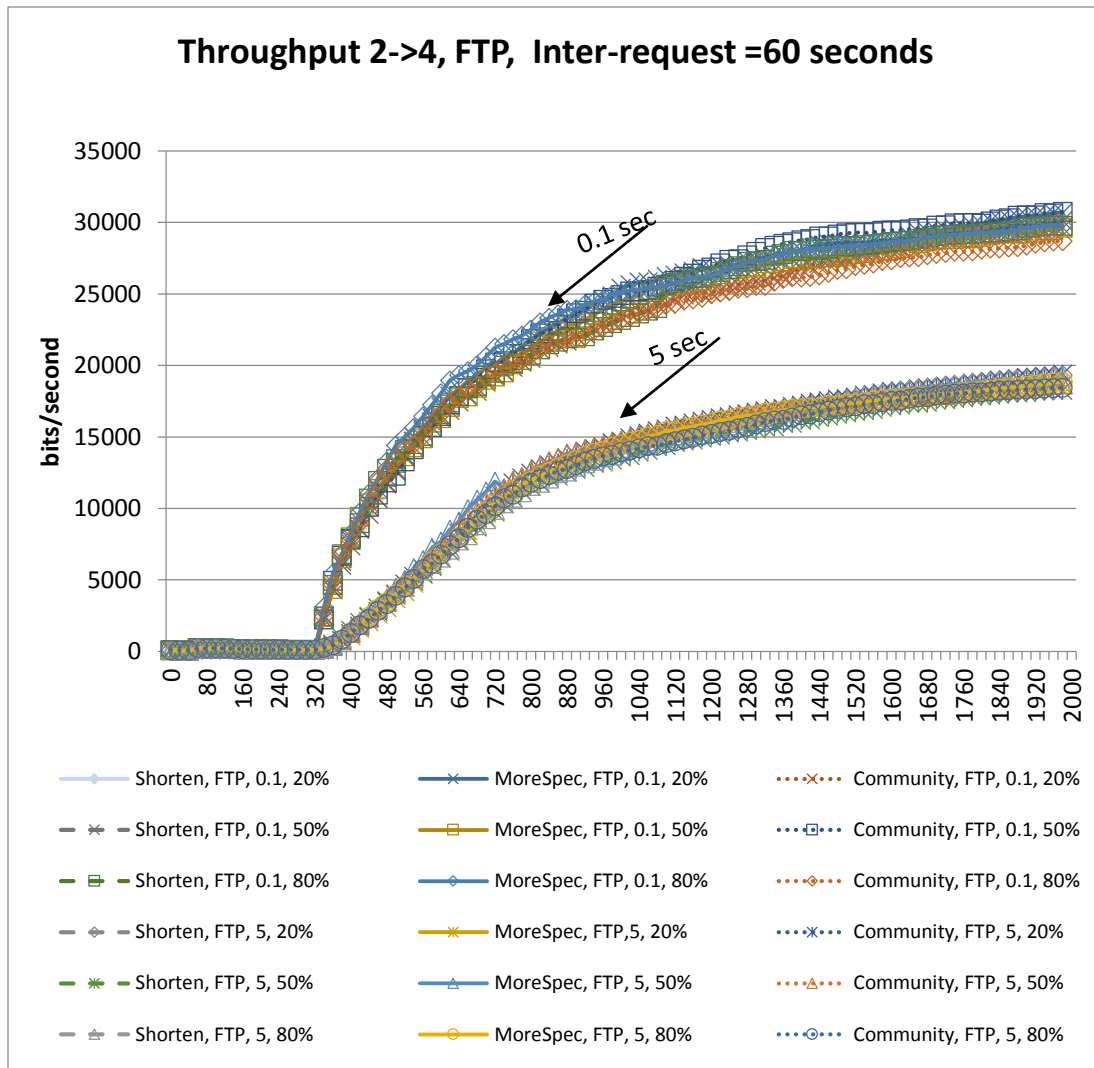
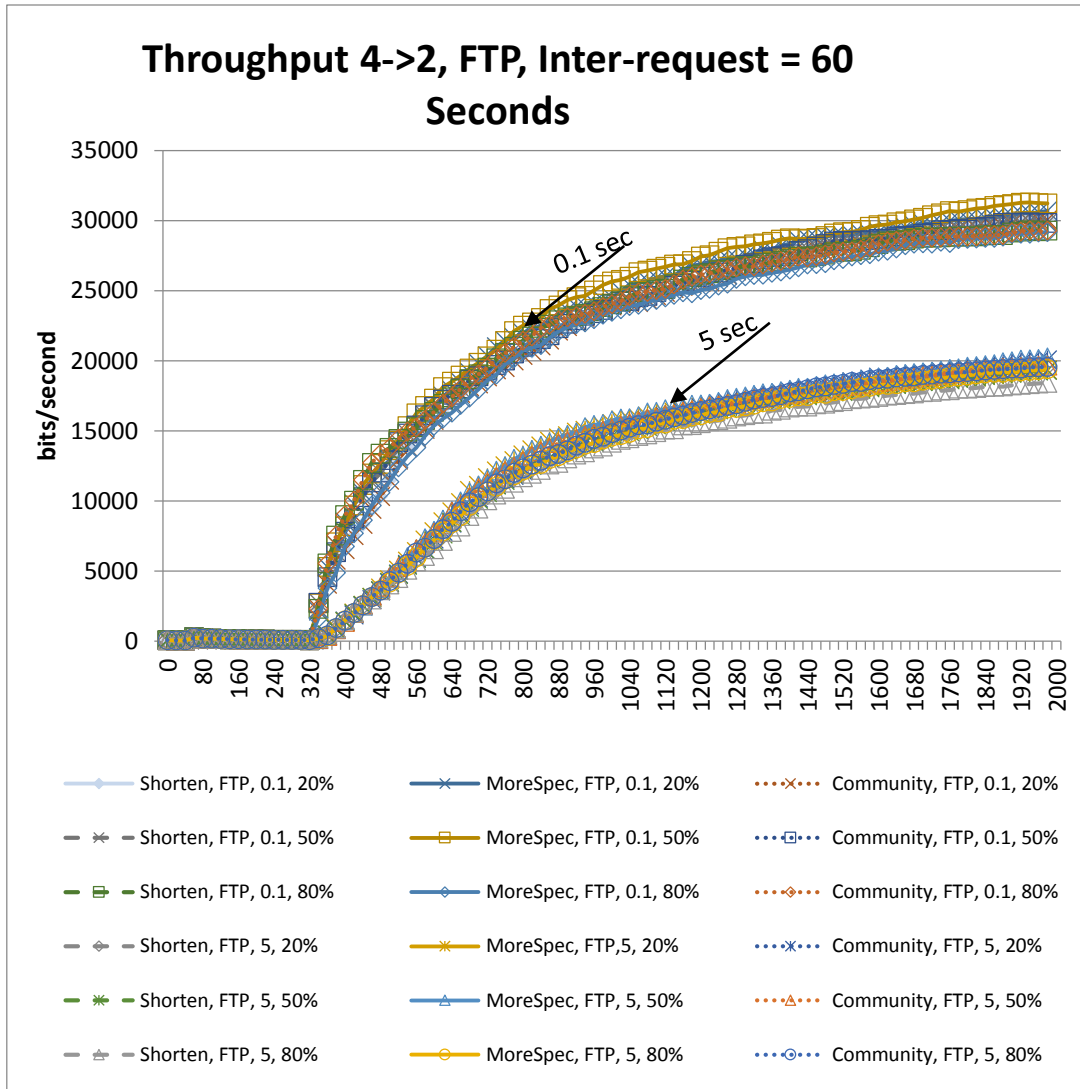


Figure D.3 Throughput for FTP application , inter-request is 60 seconds.

Figure D.4 shows the FTP throughput between Router 4 and Router 2 when the inter-request time is 60 seconds.



[Figure D.4 Throughput for FTP application , inter-request is 60 seconds.]

References

- [1] A. Alrefai, "BGP based Solution for International ISP Blocking,". MS Thesis submitted to Deanship of Graduate Studies, King Fahd University of Petroleum and Minerals, December 2009.
- [2] BGP RFC 4271: "A Border Gateway Protocol 4 (BGP-4)" Jan 2006 available at: <http://tools.ietf.org/html/rfc4271>.
- [3] Y. Rekhter, T. Li, and S. Hares. (2006, Jan.) IETF-A Border Gateway Protocol 4 (BGP-4).[Online]. www.ietf.org/rfc/rfc4271.txt
- [4] Geng-Sheng Kuo; Shu, K.C.; , "Design of global hierarchical routing architecture on future IPv6 Internet," Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE , vol.1, , pp.121-125 vol.1, 2001doi: 10.1109/GLOCOM.2001.965091
<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=965091&isnumber=20832>
- [5] OPNET: www.opnet.com.
- [6] D. Goldenberg, L. Qiu, H. Xie, Y. R. Yang, and Y. Zhang:"Optimizing Cost and Performance for Multihoming" In Proc. ACM SIGCOMM, August 2004.
- [7] K. Butler, T. Farley, P. McDaniel, and J. Rexford. "A survey of BGP security issues and solutions". Proceedings of the IEEE, vol. 98, pp. 100-122, January 2010.
- [8] Tatipamula, M.; Grossetete, P.; Esaki, H. "IPv6 integration and coexistence strategies for next-generation networks", Communications Magazine, IEEE, On page(s): 88 - 96 Volume: 42, Issue: 1, Jan 2004.

- [9] R. Barrett, S. Haar, and R. Whitestone, "Routing snafu causes Internet outage Interactive Week, Apr. 25, 1997.
- [10] O. Nordstrom, C. Dovrolis, "Beware of BGP attacks," in ACM SIGCOMM CCR, April 2004.
- [11] Y. Hu, A. Perrig, and D. Johnson, "Efficient security mechanisms for routing protocols," in Proc. ISOC Network and Distributed Systems Security Symp. (NDSS), San Diego, CA, Feb. 2003.
- [12] S. Kim, H. Lee, and Y. W. Lee, "Improving Resiliency of Network Topology with Enhanced Evolving Strategies," in Proceedings of the Sixth IEEE International Conference on Computer and Information Technology 2006, p. 149.
- [13] R. Cohen, K. Erez, D. ben-Avraham, and S. Havlin, "Breakdown of the internet under intentional attack," *Phy Rev Lett*, vol. 86, pp. 3682-3685, Apr. 2001.
- [14] D. Dolev, S. Jamin, O. Mokryn, and Y. Shavitt, "Internet resiliency to attacks and failures under BGP policy routing," *Computer Networks: The International Journal of Computer and Telecommunications Networking* vol. 50, pp. 3183 - 3196 Nov. 2006.
- [15] C. Jin, H. Wang, and K. G. Shin, "Hop-count filtering: An effective defense against spoofed DDoS traffic," in Proc. Of the 10th ACM Conference on Computer and Communications Security, 2003
- [16] X. Liu and L. Xiao, "A Survey of Multihoming Technology in Stub Networks: Current Research and Open Issues," in *IEEE Network Magazine*, May/Jun 2007.

[17] Pekka Savola: IPv6 Site Multihoming Using a Host-based Shim Layer. ICN/ICONS/MCL 2006: 50.

[18] Network Simulator (NS), <http://www.nsnam.org/>.

[19] Qiang Li; Tao Qin; Xiaohong Guan; Qinghua Zheng, "Empirical analysis and comparison of IPv4-IPv6 traffic: A case study on the campus network," Networks (ICON), 2012 18th IEEE International Conference on , vol., no., pp.395,399, 12-14 Dec. 2012 doi: 10.1109/ICON.2012.6506590

|

VITA

Name : [Mohammed Abdullah Omer Al-Mehdhar]

Nationality : [Yemeni]

Date of Birth : 31/10/1983

Email m.almehdhar@gmail.com

Address : [AL-Mukalla, Hdhrmout, Yemen..]

Academic Background : [Holder of BS in Computer Science from Hdhrmout University for Science and Technology [AL-Mukalla-Yemen]. Completed MS degree requirements in computer Networks from King Fahd University of Petroleum and Minerals [Dhahran –Saudi Arabia].(2013). My fruitful Master’s program was characterized by a number of research achievements. My thesis, focused on three major issues: how to provide a highly resiliency in the next generation of the Internet using IPv6 , how to test some approaches to solve the problem of intentional internet blocking in IPv6 , and customizing Opnet simulator and making some code modification of the Opnet to support IPv6 real environment features . Also , I publish journal with Dr. Talal Khorubi titled “COMPREHENSIVE COMPARISON OF VOIP SIP PROTOCOL PROBLEMS AND CISCO VOIP SYSTEM” paper and two other papers , one under reviewing and the other is in editing phase . Moreover, I’m a member of Saudi Honey net chapter at KFUPM and did multiple presentations and researches belongs to the Honeynet project. Although I am fairly familiar with Opnet simulator modeling as well as coding and most of the Honeynet tools, Cisco configurations and network administration and programming languages such as C,C++,VB and C#.

|